CONTENTS

Page

FOREWORD . . . . iii 1/A5

1. ON THE CONGRUENT FEATURE IN NASTRAN 1 1/A9
by P. R. Pamidi (Computer Sciences Corporation)

2. DEVELOPMENT OF THE LEARJET 28/29 WING USING
NASTRAN ANALYSIS. 11 1/B6
by Robert J. Boroughs (Gates Learjet Corporation)

3. TENSIONING OF A BELT AROUND A DRUM USING
MEMBRANE ELEMENTS 33 1/E1
by C. H. S. Chen (B. F. Goodrich Company)

4. THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS
SUBJECTED TO ASYMMETRIC THERMAL LOADS 41 1/E12
by Gordon R. Negaard (ANAMET Laboratories, Inc.)

5. STRESS CONCENTRATIONS IN SCREW THREADS. 65 2/B1
by G. Peter O'Hara (U. S. Army Armament Research and
Development Command)

6. CONDENSING LOADED POINTS FOR TRANSIENT BY
SUBSTRUCTURING. 79 2/C6
by Thomas G. Butler (Butler Analysis)

7. IMPLEMENTATIONS OF A TRAPEZOIDAL RING ELEMENT IN
NASTRAN FOR ELASTIC-PLASTIC ANALYSIS. 101 2/E5
by P. C. T. Chen and G. P. O'Hara (U. S. Army
Armament Research and Development Command)

8. NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD
IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT 113 2/F5
by Karl A. Meyer (Planning Research Corporation)

9. STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED
PERIODIC STRUCTURES 131 2/G9
by Anthony J. Kalinowski (Naval Underwater Systems
Center)

10. APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS
AT NORTHROP 165 3/C4
by Ashok K. Singh (Northrop Corporation)

11. IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS
OPERATING SYSTEM. 189 3/F2
by Stephen S. Britten and Betsy Schumacker (M.I.T.)

MICROFILMED FROM BEST
AVAILABLE COPY

Page

12. IFEMS - AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM 205
by Spencer McKellip, Todd Schuman, and Spencer Lauer (Sikorsky Aircraft) 3/G5

13. APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL. 223
by James L. Rogers, Jr. (NASA Langley Research Center) 4/B6

JUN 1 7 1980 A-830-H-10 NAS 1.55: 2131

COMPLETED

NASA Conference Publication 2131

Eighth NASTRAN® Users' Colloquium

ORIGINAL

Proceedings of a Colloquium
held at Goddard Space Flight Center
Greenbelt, Maryland
October 30-31, 1979

MICROFILMED FROM
BEST AVAILABLE COPY

NASA

250

NASA Conference Publication 2131

Eighth NASTRAN® Users' Colloquium

Proceedings of a Colloquium
held at Goddard Space Flight Center
Greenbelt, Maryland
October 30-31, 1979

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1980

FOREWORD

NASTRAN (NASA STRUCTURAL ANALYSIS) is a large, comprehensive, nonproprietary, general purpose finite element computer code for structural analysis which was developed under NASA sponsorship and became available to the public in late 1970. It can be obtained through COSMIC (Computer Software Management and Information Center), Athens, Georgia, and is widely used by NASA, other government agencies, and industry.

NASA currently provides continuing maintenance of NASTRAN through COSMIC. Because of the widespread interest in NASTRAN, and finite element methods in general, the Eighth NASTRAN Users' Colloquium was organized and held at the Goddard Space Flight Center, October 30-31, 1979. (Papers from previous colloquia held in 1971, 1972, 1973, 1975, 1976, 1977, and 1978 are published in NASA Technical Memorandums X-2378, X-2637, X-2893, X-3278, X-3428, and NASA Conference Publications 2018 and 2062.) The Eighth Colloquium provides some comprehensive general papers on the application of finite element methods in engineering, comparisons with other approaches, unique applications, pre- and post-processing or auxiliary programs, and new methods of analysis with NASTRAN.

Individuals actively engaged in the use of finite elements or NASTRAN were invited to prepare papers for presentation at the colloquium. These papers are included in this volume. No editorial review was provided by NASA or COSMIC, however, detailed instructions were provided each author to achieve reasonably consistent paper format and content. The opinions and data presented are the sole responsibility of the authors and their respective organizations.

Cochairmen:

Robert L. Brugh
NASTRAN Project Manager
COSMIC
University of Georgia
Athens, Georgia

and

Reginald S. Mitchell
Goddard Space Flight Center
Greenbelt, Maryland

BLANK PAGE

CONTENTS

Page

FOREWORD iii

1. ON THE CONGRUENT FEATURE IN NASTRAN 1
by P. R. Pamidi (Computer Sciences Corporation)

2. DEVELOPMENT OF THE LEARJET 28/29 WING USING NASTRAN ANALYSIS. 11
by Robert J. Boroughs (Gates Learjet Corporation)

3. TENSIONING OF A BELT AROUND A DRUM USING MEMBRANE ELEMENTS 33
by C. H. S. Chen (B. F. Goodrich Company)

4. THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS SUBJECTED TO ASYMMETRIC THERMAL LOADS 41
by Gordon R. Negaard (ANAMET Laboratories, Inc.)

5. STRESS CONCENTRATIONS IN SCREW THREADS. 65
by G. Peter O'Hara (U. S. Army Armament Research and Development Command)

6. CONDENSING LOADED POINTS FOR TRANSIENT BY SUBSTRUCTURING. 79
by Thomas G. Butler (Butler Analysis)

7. IMPLEMENTATIONS OF A TRAPEZOIDAL RING ELEMENT IN NASTRAN FOR ELASTIC-PLASTIC ANALYSIS. 101
by P. C. T. Chen and G. P. O'Hara (U. S. Army Armament Research and Development Command)

8. NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT 113
by Karl A. Meyer (Planning Research Corporation)

9. STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED PERIODIC STRUCTURES 131
by Anthony J. Kalinowski (Naval Underwater Systems Center)

10. APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS AT NORTHROP 165
by Ashok K. Singh (Northrop Corporation)

11. IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS OPERATING SYSTEM. 189
by Stephen S. Britten and Betsy Schumacker (M.I.T.)

Page

12. IFEMS - AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM 205
by Spencer McKellip, Todd Schuman, and Spencer Lauer (Sikorsky Aircraft)

13. APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL. 223
by James L. Rogers, Jr. (NASA Langley Research Center)

ON THE CONGRUENT FEATURE IN NASTRAN

P. R. Pamidi
Computer Sciences Corporation
Silver Spring, Maryland

ABSTRACT

The congruent feature, which is a capability in NASTRAN that can contribute to significant increases in computational efficiencies, is discussed in this paper. The usage of the capability and the software design characteristics affecting it are explained. The factors affecting the efficiency of the feature are pointed out. The details pertaining to the software design of the congruent feature are presented; in particular, the congruent element table is described. Several examples employing the congruent feature are considered and comparisons of EMG (Element Matrix Generator) module CPU times with and without this feature are presented. The results of the paper clearly demonstrate the role of the congruent feature in increasing computational efficiencies and its applicability to large-size problems.

INTRODUCTION

An important step in any NASTRAN problem is the generation of element matrices (stiffness, mass, and damping matrices as required) in the EMG module. In many cases, this step can represent a significant portion of the total problem activity. Because of the differences in algorithms and procedures, the cost of generating the element matrices for an element depends on the element type, its configuration and its properties. However, this cost is associated primarily with CPU activity and is not significantly affected by core size or I/O transfers (Reference 1).

Normally, the element matrices are generated in the EMG module once for each element in the model. However, when two or more elements in the model have the same element matrices, there is no reason why the same matrices should be computed separately for each such identical element. By declaring such elements as congruent, it is possible to cause their element matrices to be computed only once for all elements in the congruent set instead of their being computed repeatedly for each of the individual elements in the set. This results, in general, in a saving of CPU time in the EMG module. In many cases, judicious formulation of the problem to facilitate the use of the congruent feature can result in substantial savings in the computational effort. In some problems, over 90 percent reductions in EMG module CPU times have been obtained.

The congruent feature is not yet adequately publicized in the NASTRAN documentation. Currently, it is only referenced in the NASTRAN User's Manual (Reference 2) with a one-page description of the CNGRNT bulk data card. It is hoped that the discussion of this feature herein will lead to more widespread use of this capability in large-size problems thus resulting in significant increases in computational efficiencies.

CONGRUENT FEATURE USAGE

The congruent feature is specified in NASTRAN by means of one or more CNGRNT cards in the Bulk Data Deck (Reference 2). Any number of such cards may be employed.

The CNGRNT bulk data card is an open-ended card and requires the specification of a primary element identification number and one or more secondary element identification numbers. (A description of the CNGRNT card is given in the Appendix.) The terms primary and secondary as used with regard to congruent data are purely relative and have no real significance. Generally, the primary element is the lowest numbered element in the congruent set, but this need not be so. The element matrices are computed in the EMG module only for the lowest numbered element in a congruent set (even though this element may not be the primary element). The element matrices for the rest of the elements in the congruent set are then derived from these computed matrices.

SOFTWARE DESIGN CHARACTERISTICS AFFECTING CONGRUENT FEATURE USAGE

When using CNGRNT cards, the user should be aware of the following important characteristics of the congruent capability software design in NASTRAN.

- User Responsibility for Congruency Specification

The elements declared as congruent must have characteristics (such as their orientation and geometry) that cause their element matrices in the global coordinate system to be truly identical. The program cannot test the validity of this structural specification. It is, therefore, the user's responsibility to ensure that element congruence specifications are valid. Improper congruence specifications will result in an improper structure definition and will in turn lead to erroneous results. It should be emphasized that the proper use of the congruent feature will not cause the answers to be any different from those obtained without the use of the feature, but will result in a saving of CPU time in the EMG module.

- Flexibility in Specifying Congruencies

Clearly, congruency by its very definition can apply only to elements of the same type. Thus, for instance, a bar element can be congruent only to another bar element and not to a plate element. However, because of the effective manner in which the congruent feature has been incorporated into NASTRAN (as will be evident from the discussion in a later section), elements of different types can be specified on the same logical CNGRNT card without in any way making the different element types congruent. Thus, on the same logical CNGRNT card, several bar elements can be declared as belonging to a congruent set and several plate elements can be specified as belonging to a separate congruent set. However, the user should ensure that such specifications do not lead to erroneous declarations when elements of different types have the same identification numbers.

- Provision of "Phantom" Element Identification Numbers

As a corollary to the above, it may be noted that the element identification numbers (primary or secondary) specified on a CNGRNT card need not all exist in a model. This facilitates the use of the THRU option on the card more often than is possible in many other similar cases.

- Primary Element Specification

The same element can appear as the primary ID on more than one CNGRNT card, but an element listed as a primary ID on one CNGRNT card cannot be listed as a secondary ID on another CNGRNT card. However, if a primary ID is also listed as a secondary ID on the same card, then such secondary IDs are ignored.

- Secondary Element Specification

The same secondary ID cannot be listed as congruent to two or more different primary IDs.

- Redundant Specifications

Redundant specifications on CNGRNT cards are ignored.

FACTORS AFFECTING CONGRUENT FEATURE EFFICIENCY

As indicated earlier, the use of the congruent feature results in increased computational efficiency. The degree of efficiency obtained depends on the following factors some of which can be influenced by the user input specifications:

- Number of Congruent Elements

Clearly, the larger the number of elements in a congruent set and the larger the number of sets, the higher the savings in CPU time.

- Type of Elements Specified as Congruent

Larger savings in CPU time are obtained for certain element types than for other element types. Thus, for instance, declaring two IHEX3 elements as congruent will result in more savings than declaring two IHEX1 elements as congruent.

- Type of Analysis

For a specified congruent set, larger savings are obtained in dynamic analysis than in static analysis since, in the former, mass and/or damping matrices need to be computed in addition to stiffness matrices.

- Numbering of Grid Points of the Congruent Elements

Processing is slightly more efficient if the relative order of the numbering of the grid points of the congruent elements is the same. Thus, for instance, two congruent quadrilateral plate elements are processed more efficiently if their grid points are numbered 1-7-4-6 and 12-23-16-20, respectively, than if they were numbered 1-7-4-6 and 11-14-17-15, respectively. In the former case, the grid point numbers of the two congruent elements increase or decrease in the same order as we go around the elements. In the latter case, the grid point numbers of the two congruent elements increase or decrease in different orders as we go around the elements.

SOFTWARE DESIGN OF THE CONGRUENT FEATURE

The preliminary checking of the validity of the data on the CNGRNT bulk data cards is performed in subroutine IFS1P of the IFP module, but the detailed processing of these cards is done in subroutine EMGCNG of the EMG module. Besides checking for various errors in the CNGRNT data, the EMGCNG routine sets up a table of congruent element IDs in open core. This important table forms the basis for handling congruent elements subsequently in subroutines EMGPRO and EMGOUT of the EMG module. It is, therefore, useful to know the manner in which this table is set up. (The detailed manner in which congruent elements are handled in the EMG module can be ascertained from the source code and from Reference 3.)

The congruent element table consists of a pair of words for each element (primary or secondary) specified on a CNGRNT card. The first word of this pair contains the user-specified ID of the element. The second word of the pair indicates whether the element identified by the first word is the primary ID of that congruent set or is a secondary ID of that set. In the case of a secondary ID, the second word of the pair is a positive integer specifying the open core address of its primary ID. In the case of a primary ID, the second word of the pair is either zero or a negative integer. Initially, this word is set to zero in the case of all primary IDs. When the element matrices for the first element (which is also the lowest numbered element) in a congruent set are computed, the zero in the corresponding word is changed to the negated open core address of the element matrix (or dictionary) data.

The second word of any pair of words in the congruent element table thus contains very important information. If it is a positive integer, then it is a pointer to the primary ID of that congruent set. If it is zero, then the corresponding ID in the first word is the primary ID and the element matrices have not yet been computed for that set. If it is a negative integer, then the corresponding ID in the first word is the primary ID and the element matrices have been computed for that set and can be obtained from the open core address information represented by that negative integer. The table as set up by the EMGCNG routine is sorted on the element IDs in the first words of the word pairs. An example of a congruent element table is shown in Table 1.

The unique design of the congruent element table allows for a very efficient processing of the congruent data by the EMG module. It should be noted that the EMG module will never mix element matrices for different element types. The EMG module processes each element type one after another. When it completes the processing of an element type, the negated open core addresses in the congruent element table (if any) are replaced by zeroes. Thus, when the processing of the next element type starts, the congruent element table (if any) has no history or evidence of the processing of the previous element type. Note also that the design of the table permits the specification of non-existent element IDs in the CNGRNT data.

EXAMPLES OF CONGRUENT FEATURE USAGE

There are 82 demonstration problems in Level 17.5 of NASTRAN. The congruent feature is employed in fifteen (15) of these problems. A comparison of the EMG module CPU times (on IBM S/360-95 computer) for these problems with and without the congruent feature is presented in Table 2. The savings resulting from the use of the congruent capability are quite apparent from this table. The most dramatic savings are obtained in NASTRAN Demonstration Problem Nos. 3-1-2 and 8-1-2 (UMF Problem ID Nos. 30120 and 80120, respectively) in which the EMG module CPU times are reduced by more than 99 percent.

SUMMARY

The congruent feature in NASTRAN is explained and the software design characteristics affecting its usage and the factors affecting its efficiency are discussed. The details pertaining to the software design of the capability are presented. Examples illustrating the usage of the feature are considered. The results of the paper clearly demonstrate the role of the congruent feature in increasing computational efficiencies and its applicability to large-size problems.

REFERENCES

1. Field, E. I., Herting, D. N., and Morgan, M. J., NASTRAN User's Guide, (Level 17.5), NASA Contractor Report 3146, June 1979, p. 14.3-3.

2. The NASTRAN User's Manual, (Level 17.5), NASA SP-222 (05), December 1978, p. 2.4-36a.

3. The NASTRAN Programmer's Manual, (Level 17.0), NASA SP-223 (04), December 1977, Section 4.124.

APPENDIX

Input Data Card CNGRNT Identical Elements Indicator

Description: Designates secondary element(s) identical to a primary element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CNGRNT	PRID	SECID1	SECID2	SECID3	SECID4	SECID5	SECID6	SECID7	abc
CNGRNT	11	2	17	34	35	36			

+bc	SECID8	SECID9		-etc.-					

Alternate Form

CNGRNT	PRID	SECID1	"THRU"	SECID2					
CNGRNT	7	10	THRU	55					

Field	Contents
PRID	Identification number of the primary element (not necessarily the lowest number)
SECIDi	Identification number(s) of secondary element(s) whose matrices will be identical (or congruent) to those of the primary element.

Remarks:

1. Orientation, geometry, etc. must be truly identical such that the same stiffness, mass and damping matrices are generated in the global coordinate system.
2. This feature is automatically used by the INPUT module.
3. The CNGRNT feature cannot be used when an AXIC card is present in the bulk data deck.
4. An element that has been listed as a primary ID on a CNGRNT card cannot be listed as a secondary ID on another CNGRNT card. However, if the element is listed as a secondary ID on the same card, then such secondary IDs are ignored.
5. The same secondary IDs cannot be listed as congruent to two or more different primary IDs.
6. Redundant specifications on CNGRNT cards are ignored.
7. The stiffness, mass and damping matrices are actually calculated for the lowest numbered element in the congruent set (even though this element may not be the primary ID).

TABLE 1. - EXAMPLE OF A CONGRUENT ELEMENT TABLE

Open Core Location	Table Column 1	Table Column 2
Z (53)	1	54
Z (54)	3	0
Z (55)	7	59
Z (56)	12	54
Z (57)	15	59
Z (58)	18	54
Z (59)	36	0
Z (60)	40	59
Z (61)	69	54

Note: The above table represents the congruent element table as initially set up by EMGCNG routine resulting from the processing of two CNGRNT bulk data cards--one with a primary ID of 3 and secondary IDs of 1, 12, 18 and 69 and the other with a primary ID of 36 and secondary IDs of 7, 15 and 40.

TABLE 2. - EXAMPLES OF CONGRUENT FEATURE USAGE IN LEVEL 17.5 NASTRAN DEMONSTRATION PROBLEMS

Example No.	Demonstration Problem No.	UMF Problem ID No.	Congruent Element Data			EMG Module CPU Times (sec.)*		Saving in EMG Module CPU Time Obtained by Using the Congruent Feature (%) $\frac{(b) - (a)}{(b)} \times 100$
			Element Type	Number of Elements	Number of CNGRNT Sets	Using the Congruent Feature (a)	Without Using the Congruent Feature (b)	
1	1-3-1	10310	QDMEM	216	1	0.8	8.3	90.4
2	1-3-2	10320	QDMEM1	216	1	1.2	13.5	91.1
3	1-3-3	10330	QDMEM2	216	1	1.5	11.1	86.5
4	1-8-1	10810	HEXA1	40	1	0.1	3.5	97.1
5	1-9-1	10910	HEXA2	40	1	0.3	7.4	95.9
6	1-11-1	11110	QUAD1	50	1	0.2	7.7	97.4
7	1-13-1	11310	IHEX1	40	5	2.8	16.9	83.4
8	1-13-2	11320	IHEX2	2	1	2.7	4.5	40.0
9	3-1-1	30110	QUAD1	200	1	0.4	15.4	97.4
10	3-1-2	30120	QUAD1	800	1	0.8	130.5	99.4
11	5-1-1	50110	TRIA1	80	4	0.7	11.7	94.0
12	8-1-1	80110	QUAD1	100	1	0.4	5.8	93.1
13	8-1-2	80120	QUAD1	400	1	0.4	49.1	99.2
14	14-1-1	140110	QUAD2	10	5	1.7	2.3	26.1
15	15-1-1	150110	BAR QUAD2	10 20	5 5	1.4	5.0	72.0

*All of the above problems were run on the IBM S/360-95 computer.

DEVELOPMENT OF THE LEARJET 28/29 WING USING NASTRAN ANALYSIS

Robert R. Boroughs
Gates Learjet Corporation

SUMMARY

A great deal of the structural development work performed on the Learjet 28/29 wing was accomplished using Nastran analysis. This included the basic sizing of primary structural members such as wing skins, wing skin splices, and spar caps, as well as the calculation of preliminary weight estimates utilizing the weight computation routine in Nastran. The eight spar redundancy of the Learjet wing made this task somewhat more complex and challenging than for the more determinate type wing structures. The discussion that follows describes some of the problems that were encountered and the solutions and methods that were used.

INTRODUCTION

The Learjet 28/29 wing was the most significantly different derivative wing both structurally and aerodynamically, since the introduction of the Learjet Model 23. This wing evolved from the earlier Model 35/36 wing and has been installed on a modified Model 25 fuselage. The most outwardly noticeable changes to the 28/29 wing from previous Learjet wings were at the wing tips. Here the two foot extension and tip tank on the Model 35/36 wing were replaced by a six foot extension and winglet on the 28/29 wing (see Figures 1 & 2). The outward appearance of the 28/29 wing in the inboard section remained the same as the 35/36 wing, and internally this section still had eight spars as does the 35/36, but this was where most of the similarity ended. The wing skin and center line splice plate thicknesses have increased, the section properties of several of the spar caps have increased, and wing skin stringers have been extended or added.

This same basic 28/29 wing configuration was later selected as the airfoil for the Learjet Model 54/55/56 series aircraft. Some growth capability was included in the 28/29 wing for the 50 series aircraft, but the complete detail structural definition was to be determined further into the 50 series project.

BACKGROUND

Approval to proceed with the development of the Learjet 28/29 aircraft was received in February of 1977, and work began on the wing structural analysis using Nastran that same month. The basic objectives established for the wing structural redesign were to obtain satisfactory margins of safety, minimize the impact on tooling, keep the weight increases as small as possible, and complete the certification on a very tight schedule. These goals were to be achieved while operating under constraints such as limited manpower availability and increasing lead times for parts and materials. Factors such as these later influenced alternatives that were chosen during the course of this project.

Initial analytical work performed on the 28/29 wing inboard section was accomplished using the 35/36 Nastran wing model described in NASA TMX-3428 (Ref. 1). This model was later updated in the outboard section with the six foot extension and winglet attachment structure. Sizing of the structural members was to be accomplished by a combination of Nastran analysis, post processor programs, and detail stress analysis. Since the results of the 35/36 Nastran wing model had correlated very closely with the experimental data from the 35/36 wing static test, the strategy for the 28/29 wing qualification was to perform limit load tests on a highly instrumented wing static test article, establish a correlation between the Nastran results and experimental data at this point, and qualify the ultimate load conditions by analysis using Nastran results for FAA certification. The advantage of this type of approach was to reduce the costs and lead time associated with a static test.

MODEL DESCRIPTION

Since the 28/29 wing was symmetrical about the aircraft center line, a half model was used. The Nastran wing model geometry was developed from the 35/36 wing contours inboard of W.S. 181.10 and from the 28/29 wing contours outboard of W.S. 181.10. The wing surface was divided into a basic mesh which was defined by the intersection of the spar caps and rib caps. This was further subdivided in the spanwise direction by breaking these bays into equal increments where possible. Structural members modeled included the spar caps, rib caps, and wing skin stringers with ROD elements, the spar webs and shear webs with SHEAR elements, and the wing skins and wing skin splices with QDMEM2 elements (See Ref. 2).

Modeling of the wing skins included the effect of sculpturing and contouring, and the lower skin reflected the stiffness of the access doors. The wing skin stringers were modeled by dividing the stringer areas in half and lumping each half as a separate element with the adjacent spar cap. This was done in order to simplify the modeling and conserve degrees of freedom. There were four different splice plates on the 28/29 wing skin. These were the upper and lower splices at W.S.0.00 and the upper and lower splices at W.S.181.10 where the six foot wing extension was attached. The finite element representation reflected the contour and taper characteristics that had been machined into these members.

The six foot extension geometry was basically an extension of the taper and contour of the inboard wing section. The inboard eight spars were continued into the six foot extension, and two additional spars were added in the trailing edge. This spar addition was incorporated to provide stiffness and an internal load path for forces developed by the winglet, since the winglet was mounted very near the trailing edge of the wing. The winglet attachment structure was modeled from W.S. 244.10 to winglet station 6.00. This structure included spars five through ten and the winglet skin in this area. ROD elements were used to model the winglet spar caps, SHEAR elements were used to model the winglet spar webs, and QDMEM2 elements were used to model the winglet skin. Beyond winglet station 6.00 the winglet structure was modeled strictly as a load fixture using Nastran BAR elements (See Fig. 3).

Constraints for the model were applied in the spanwise direction at the W.S.0.00 spar caps, and in the vertical direction at the wing attachment fittings at spars 2, 5, 7 and 8, and in the fore-aft direction at spar 5. Ten basic load conditions were examined during the static analysis. These cases consisted of positive and negative gust loads as well as various landing loads.

STRUCTURAL CONFIGURATION DEFINITION

Preliminary design analysis of the existing 35/36 wing structure had revealed that larger section properties or higher allowables would be necessary to sustain the increased 28/29 loads. During the initial phase of wing redesign strong emphasis was placed on retention and utilization of existing tooling. This was done in an attempt to keep tooling costs and the parts count down and simplify the fabrication and assembly process. Consequently, several different configurations were analyzed where the wing skin was selected as the primary member for material addition with reinforcement of the spar caps in localized areas as the secondary means of material addition. This type of approach normally has been reserved to supersonic airfoil construction where a thin wing chord section eliminates many possible structural configurations (See Ref. 3), but in this situation the constraints were more cost oriented.

Each successive configuration examined had a thicker skin than the preceding configuration, but many of the spar caps still had unacceptably high stress levels. By this time the weight increases had become substantial and the impact of this parameter on flight performance had become a serious factor. Consequently, this approach was eliminated as an acceptable solution for obtaining the basic design goals.

A new approach was then chosen where the emphasis was placed on increasing the spar cap areas in combination with the wing skin thickness as the means for developing satisfactory stress levels. Based on the wing skin studies that were conducted earlier, a wing skin configuration was selected for the 28/29 wing. The thickness of these skins were slightly greater than the existing 35/36 wing skins, but the total thickness was also considerably less than most of the other configurations previously examined. The material selected for both the upper surface and lower surface was 2014-T6. This was the same material that had been used on the upper surface of the 35/36 wing, but on the lower surface the 2014-T6 was used in place of 2024-T3. Selection of this material was influenced to a great extent by raw stock lead times in effect during that period, as well as the change in loads from the 35/36 wing to the 28/29 wing.

Using the basic wing skin selected from the previous studies, spar cap areas were increased in the regions where the margins of safety were deficient. This process initially concentrated on the wing section inboard of the landing gear rib where the stresses were the highest. When this region was improved to satisfactory levels, the process was expanded to the region outboard of the landing gear rib, and from there on out to the winglet attachment structure.

A localized buckling analysis which has been described in NASA TMX-3428 (Ref. 1) was used to determine the non-linear effect of wing skin buckling on the spar cap stresses. This analysis generally required several iterations before a convergent solution was obtained.

INTERNAL LOADS REDISTRIBUTION

Redundancy in the 28/29 wing with the multiple spar construction has some very distinct advantages for fail safe capability, but this same asset makes the structure somewhat more difficult to analyze. Nastran finite element analysis has made this task more manageable and has permitted a better understanding of this complex structure. As the first series of iterations on the inboard spar cap areas were approaching a convergent solution, there was observed a significant redistribution of internal loads from the previous configuration. As material was added to the critical sections, there appeared to be a significant redistribution of internal loads from the less critical areas into the more critical areas (See Fig. 4). Although stress levels decreased in the critical areas, these levels did not decrease linearly with the increase in spar cap area, and the stress levels in the non-critical regions also decreased at the same time. This reduction in stress level in the non-critical areas may have seemed to indicate that material could have been removed from these areas to help reduce weight, but there was obviously another factor to consider. Further area reductions in the non-critical regions would have increased the stresses in the critical regions further, and created a need for more material additions in those regions. This redistribution of internal loads into a few key structural members also raised serious questions as to whether an effective and efficient fail safe qualification could be used for a structure defined in this manner.

As a result of the concern for maintaining an effective and efficient fail safe capability for the 28/29 configuration, a different approach was selected for establishment of the spar cap section properties. This new approach emphasized maintaining an internal load and stress level balance across the chord section of the wing. This was accomplished by initially assigning equal areas to each of the spar caps from the leading edge to the trailing edge, and increasing each spar cap area by an equal increment for each iteration until a satisfactory margin of safety was achieved for the critical member. This worked out quite well and the weight penalties were not quite as severe as was seen in the first approach (See Fig. 5). After acceptable margins of safety were achieved in the critical spars, area reductions were then made on some of the less critical spars. These spars were generally located near the leading edge of the wing and the trailing edge of the wing. These spars were not generally highly loaded, and the reduction in these spar cap areas had little impact on the other spars. Further weight reductions were made by tapering the spar caps in the spanwise direction.

A new material was also chosen for the spar caps inboard of the landing gear rib in order to obtain higher allowables that were more compatible with the wing skin allowables and to also help reduce the weight of these members.

This material was 7075-T73 extrusion, and in addition to the improved allowable values this material also had improved stress corrosion resistant properties over some of the other 7000 series aluminum alloys.

Improvement of the upper surface capability outboard of the landing gear rib was another area which received considerable attention. On previous Learjets the spar caps in this region had been constructed from bent up sheet metal channels which tapered in thickness from the inboard end to the outboard end as opposed to the extruded caps attached to shear webs in the inboard section. All of the spar caps in the wing section outboard of W.S. 53 were of nearly equal areas; thus maintaining a fairly even internal loads distribution. The margins of safety for these members were generally not as deficient as the inboard spar caps, and other methods were used to correct the low margins than were used in the inboard region.

The buckling analysis that was mentioned earlier revealed that there were a number of panels that indicated advanced stages of buckling. To help relieve the spar cap stresses existing stringers that were used on the 35/36 wing were extended further into the outboard sections, and stringers were added to some bays where no stringers had existed previously. This not only caused the skins to carry more load in compression, but also added more basic area very near the outside fiber to help reduce the bending stresses on the spar caps. In some areas the use of wing skin stringers was not sufficient to obtain satisfactory stress levels and local reinforcements were added.

PRELIMINARY WEIGHT ESTIMATES

Preliminary weight estimates were arrived at with the aid of the weight calculation routine in Nastran. Two PARAM cards were inserted into the Bulk Data deck. The first PARAM card called out the GRDPNT option, and the second card called out the WTMASS feature. Accordingly, density factors were added to all of the material cards. A model 35/36 wing was run first to determine a base line weight upon which weight increases for the 28/29 wing would be determined. Although this routine does not include such detail factors as fuel sealer weights, control mechanism weights, and other miscellaneous factors, the preliminary weight estimates were still considered to be a reasonably accurate measure of weight increase over the 35/36 wing.

DOWN BENDING ANALYTICAL QUALIFICATION

Originally both the up bending and down bending ultimate load conditions were proposed to be qualified by analysis. Although there was a great deal of analytical work done on the up bending load condition, this load case was eventually qualified by static test due to the tight schedule and lack of man power availability that was prevalent at that time. However, the ultimate down bending load case was certified by analysis. The down bending load condition was not as critical as the up bending load condition, and the 28/29 wing did not require nearly as much rework for this condition as was necessary for the up bending condition. A comparison of the 28/29 down bending loads with the 35/36 down bending loads showed that the 28/29 loads were greater than the 35/36 loads, but not by a large amount. Considering the material additions to the lower wing skin thickness, there was very good reason to expect that the stress levels might be very nearly the same.

To achieve this analytical qualification a correlation was first established between the 35/36 wing static test strain levels and the 35/36 Nastran strain levels inboard of W.S. 181.10. A comparison was then made between the 35/36 Nastran strain and the 28/29 Nastran strain levels. Im almost every location the 28/29 wing Nastran strain levels were less than or equal to the 35/36 Nastran strain levels. In those areas where the 28/29 wing down bending strains exceeded the 35/36 down bending strains margins of safety were calculated, and in all cases these margins of safety were shown to be more than adequate.

The wing structure between W.S. 181.10 and 244.10 commonly referred to as the six foot extension was qualified by analysis and proof load tests. Generally the stress levels in this section were low and had quite high margins of safety. The structure outboard of W.S. 244.10 consisted entirely of the winglet and winglet attachment structure. Due to the complexity of this member, certification was accomplished with a static test for both the up bending and down bending conditions. Typical plots showing the relationship between the 35/36 wing experimental data, the 35/36 Nastran wing data, and the 28/29 Nastran wing data have been shown in Figures 6 thru 9.

STATIC TEST UP BENDING CORRELATION

There were over 400 strain gages installed on the 28/29 wing static test article. This was more than twice the number of gages installed on any previous Learjet wing test. All the strain gages were installed on the right hand wing to simplify the installation and instrumentation. During the static test these gages were monitored on a cathode ray tube (CRT) using an interactive graphics program. This program provided a quick means of monitoring the status of the static test article and identifying areas that could become critical. Upon command the program would list the top 15 gages in tension and compression on the CRT as well as on a hard copy printer. Warnings were also issued for non-linear gages above a certain strain level, and individual stress versus load plots could be obtained within seconds for any strain gage channel. Using previously calculated Nastran stresses in key areas, a comparison was made with the appropriate strain gage channel to determine whether the test was proceeding as planned. This approach proved extremely valuable in monitoring and controlling the static test (See Fig. 10).

At the conclusion of the static test the strain gage results were used for a more detailed comparison with Nastran analytical results. The correlation of this experimental data with the Nastran data was generally very good for the majority of strain gage locations. Agreement was probably best in the wing section outboard of the landing gear rib where the structure was most uniform. Correlation in the section inboard of the landing gear rib was good, but in some locations there was more noticeable deviation which appeared to be significantly influenced by structural cutouts and discontinuities in the vicinity of the gage attachment. Figure 11 shows the strain values on the spar 5 upper cap, and Figure 12 shows the strain values on the spar 5 lower cap. These values correlate quite well except in the area of the carry through fittings at wing station 53. These fittings were installed to maintain continuity of the spar caps which were interupted by the landing gear rib at this point.

Correlation of the strain gage values and the Nastran results were shown in Figure 13 for the spar 4 upper cap and in Figure 14 for the spar 4 lower cap. Again agreement was generally very good except in a couple of areas. The first area was mentioned previously in regard to spar 5 in the vicinity of the landing gear rib. The other area of some deviation was in the area adjacent to the W.S. 181.10 rib. Here the wing skins were discontinuous and were spliced by a fingered and contoured splice plate both upper and lower. The spars were also discontinuous in this region, and splices were installed for spar cap continuity.

RESULTS AND CONCLUSIONS

The Model 28/29 wing development at Learjet was significantly influenced by Nastran analysis. Configuration development and member sizing were performed much more accurately and faster than could have been done previously. This permitted Learjet to determine the impact of changes in the wing structural arrangement and member section properties on stress levels, internal loads, and aircraft weight at a much earlier point in the wing development.

Substantiation of the 28/29 wing was accomplished by a combination of testing and analysis where Nastran was the basis for much of the analytical work. During the static test phase of certification there were no structural failures in the 28/29 wing due to any of the design load conditions. Correlation of the analytical results with the experimental data was generally very good except in areas where there were discontinuities. The ultimate down bending load condition was qualified by Nastran analysis which reduced the cost and lead time for this segment of the certification.

REFERENCES

1. Abla, Mike H., Boroughs, Robert R. and Cook, Everett L. Analysis of The Learjet 35/36 Wing and Correlation with Experimental Results, pp. 331-352, Nastran User's Experiences, TMX-3428, October, 1976.

2. The Nastran User's Manual (Level 17.0), NASA SP-222(04), Washington, D.C., December, 1977.

3. Bruhn, E. F.: Analysis and Design of Flight Vehicle Structures. Tri State Offset, 1973 (pp. A19.1).

BLANK PAGE

BLANK PAGE

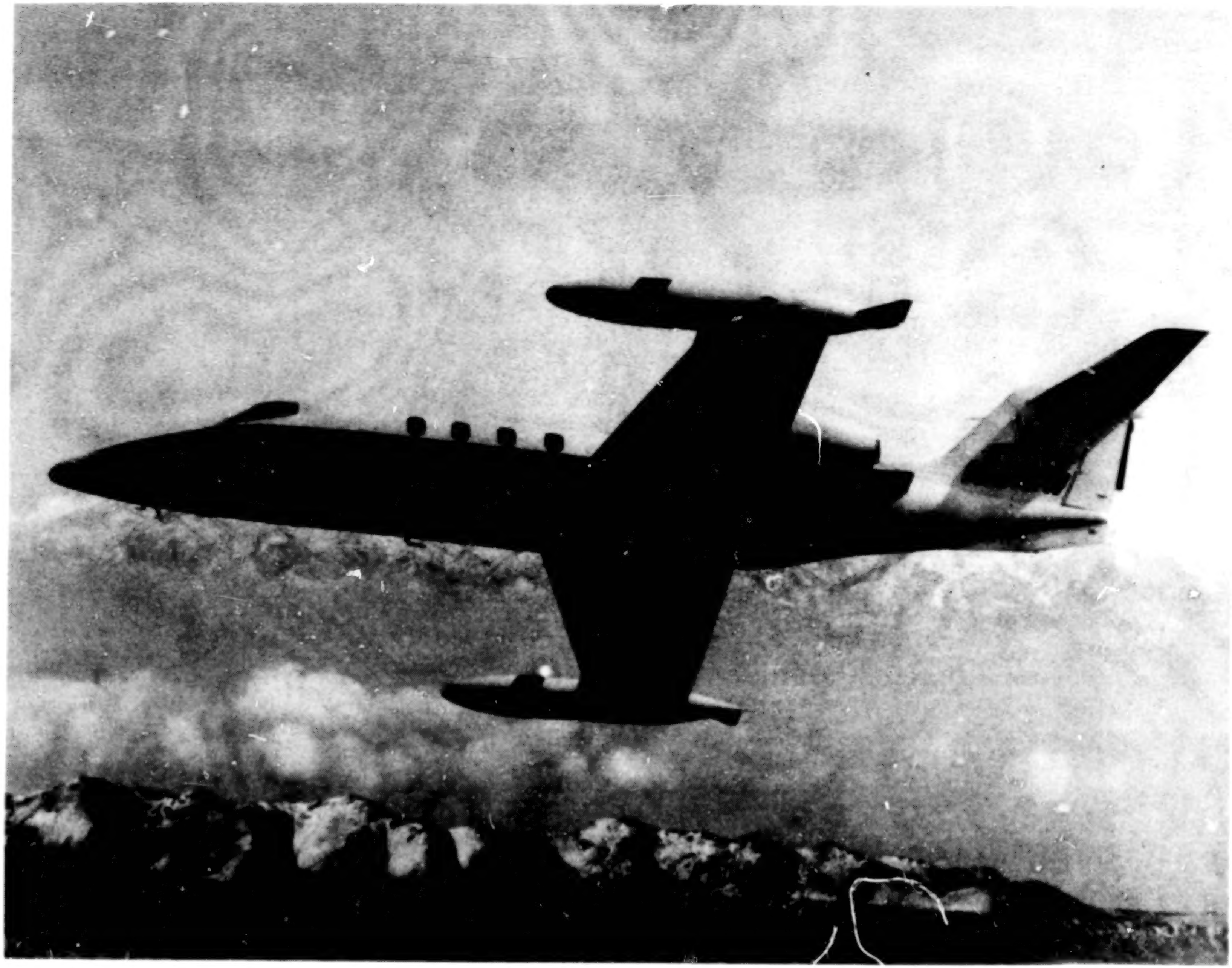


FIGURE 1 - LEARJET MODEL 35/36 AIRCRAFT

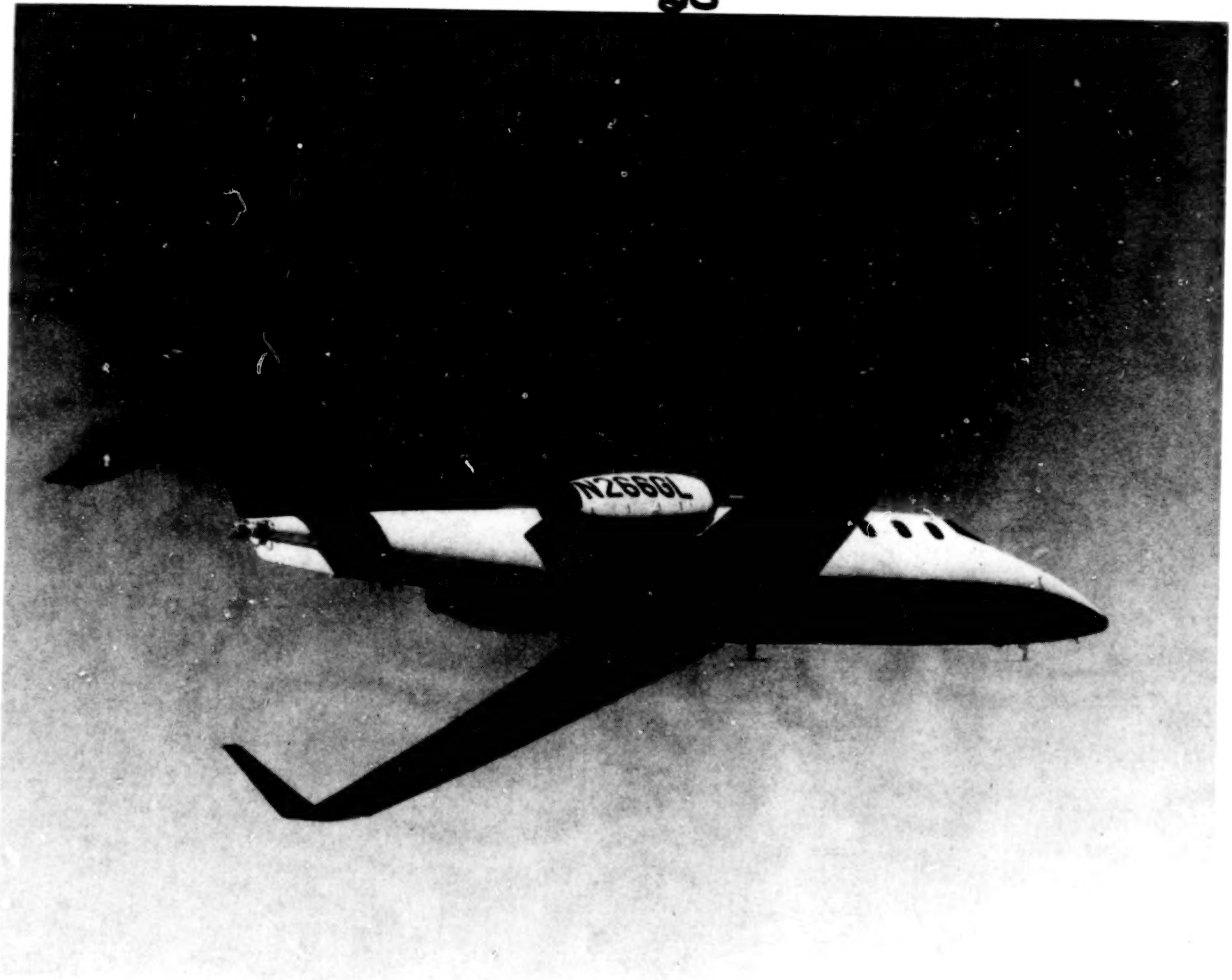


FIGURE 2 - LEARJET MODEL 28/29 AIRCRAFT

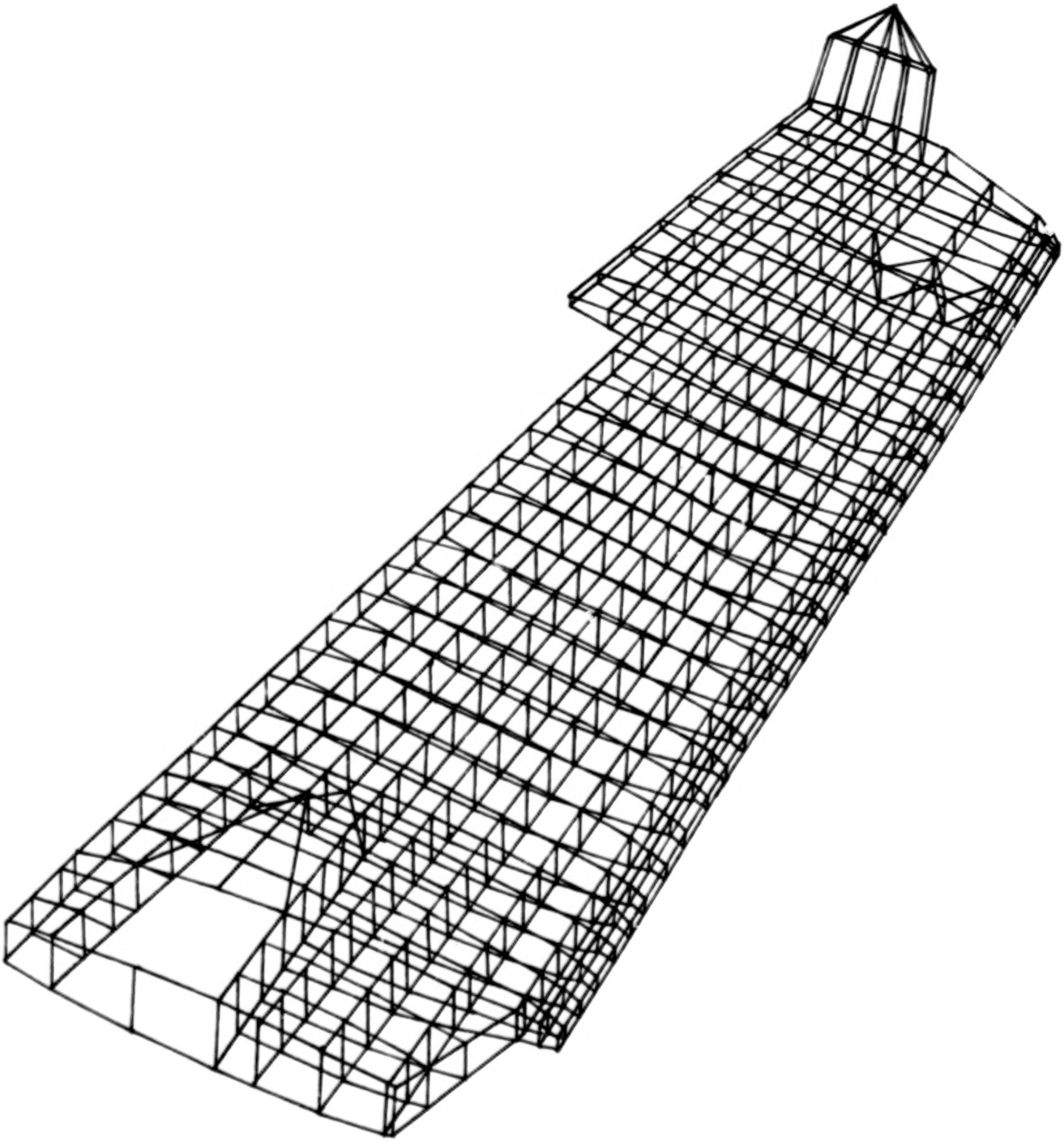


FIGURE 3 - NASTRAN 28/29 WING MODEL

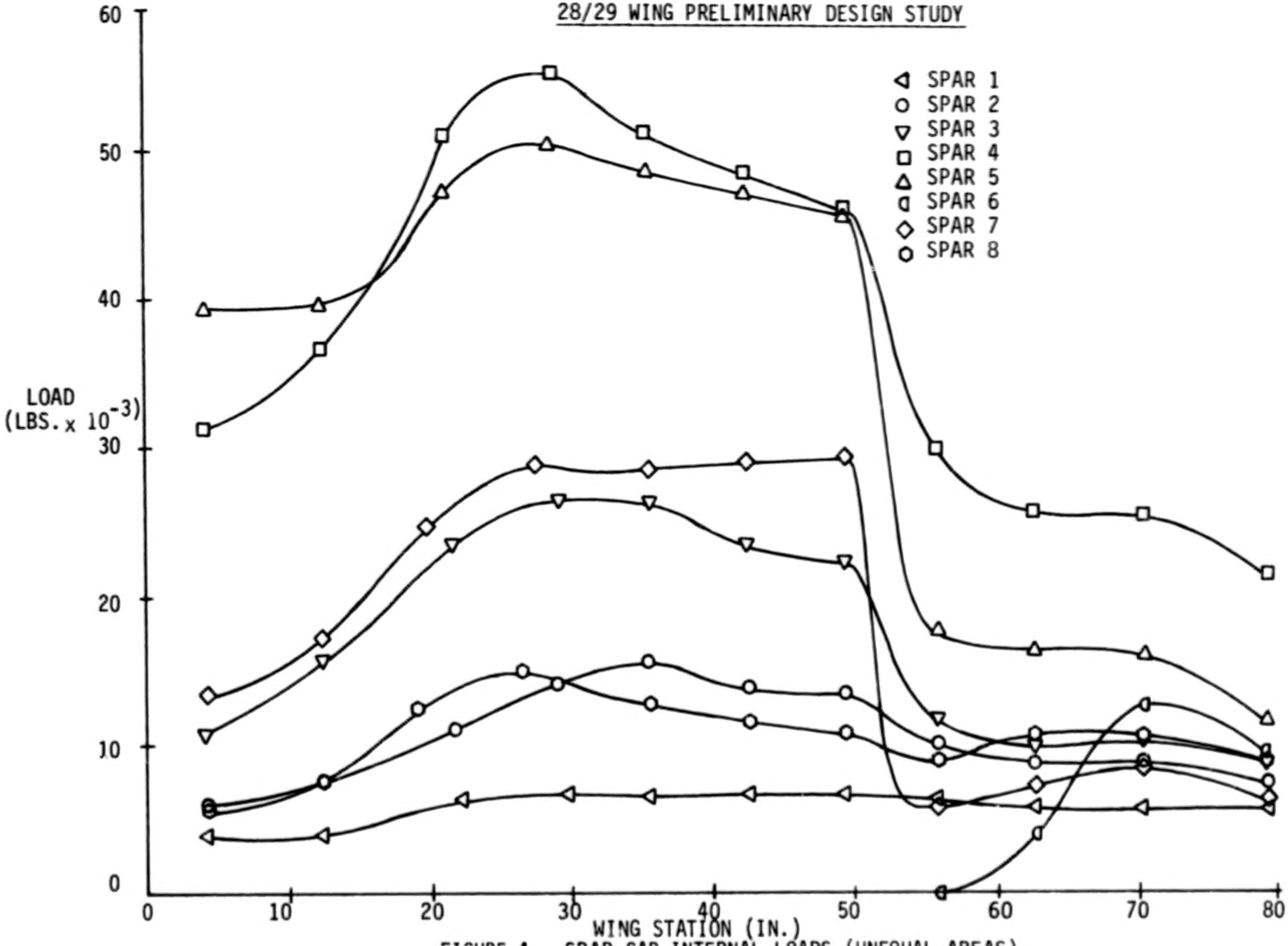


FIGURE 4 - SPAR CAP INTERNAL LOADS (UNEQUAL AREAS)

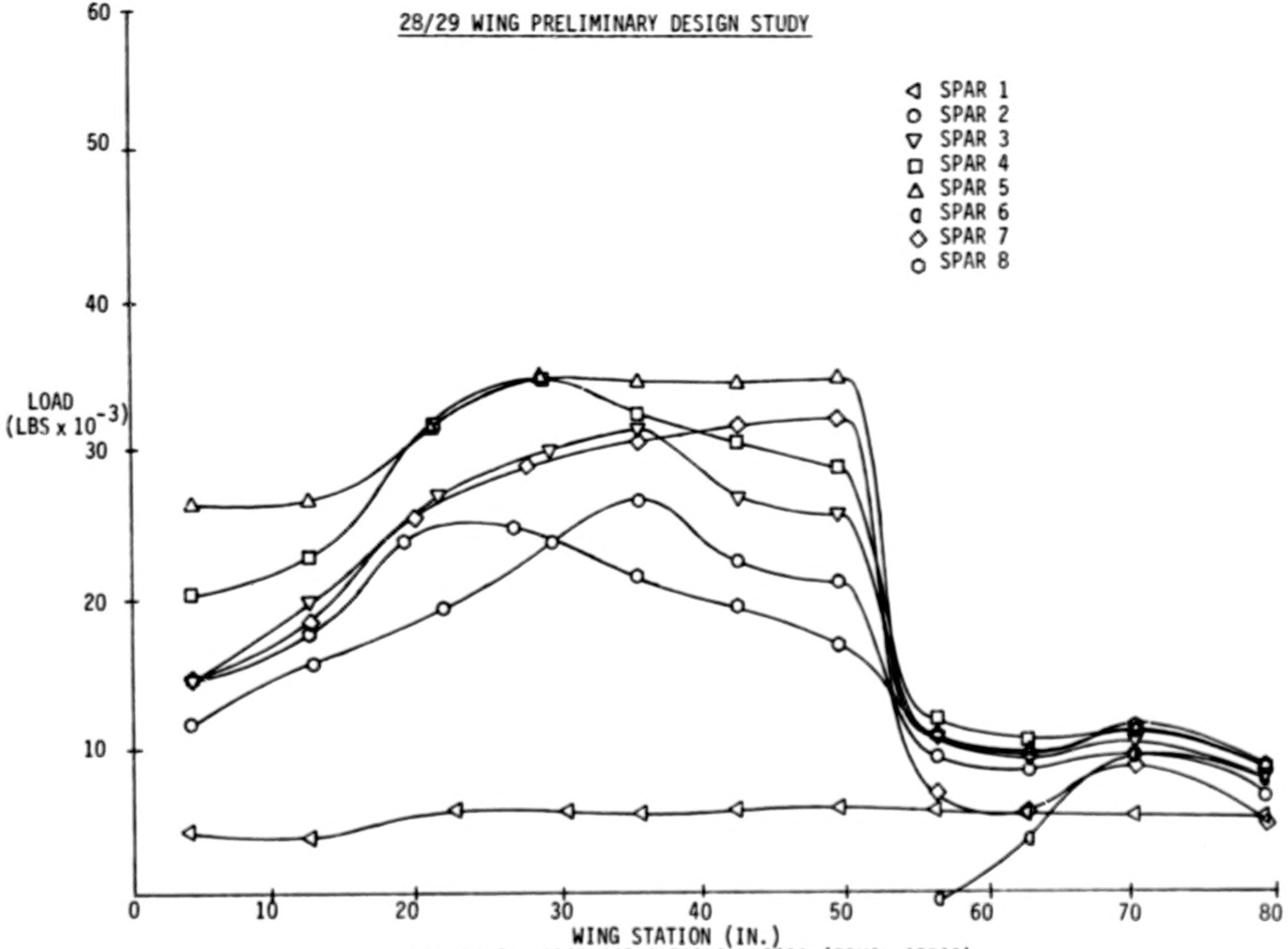


FIGURE 5 - SPAR CAP INTERNAL LOADS (EQUAL AREAS)

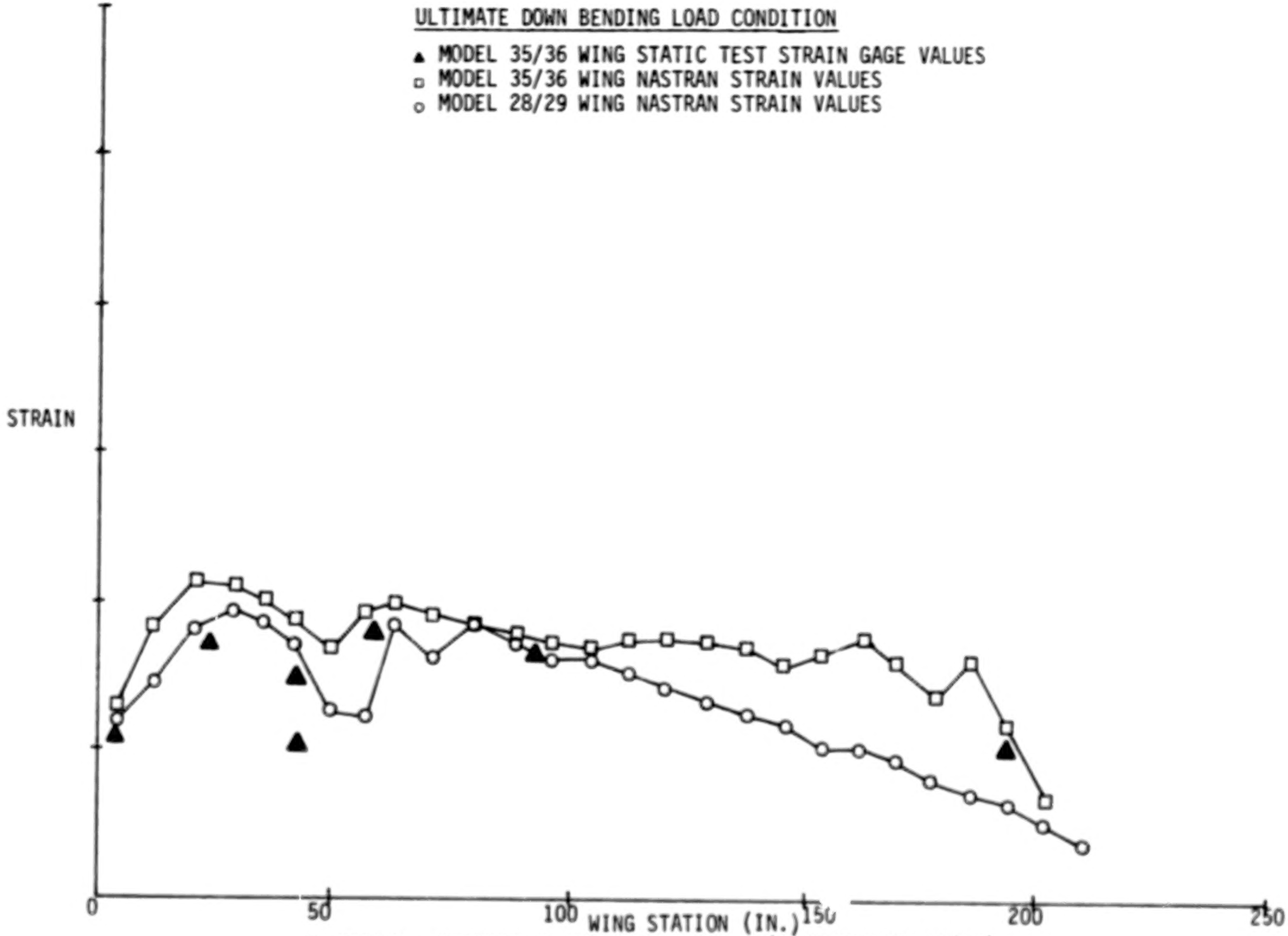


FIGURE 6 - STRAINS IN UPPER SPAR CAP 4 (DOWN BENDING LOADS)

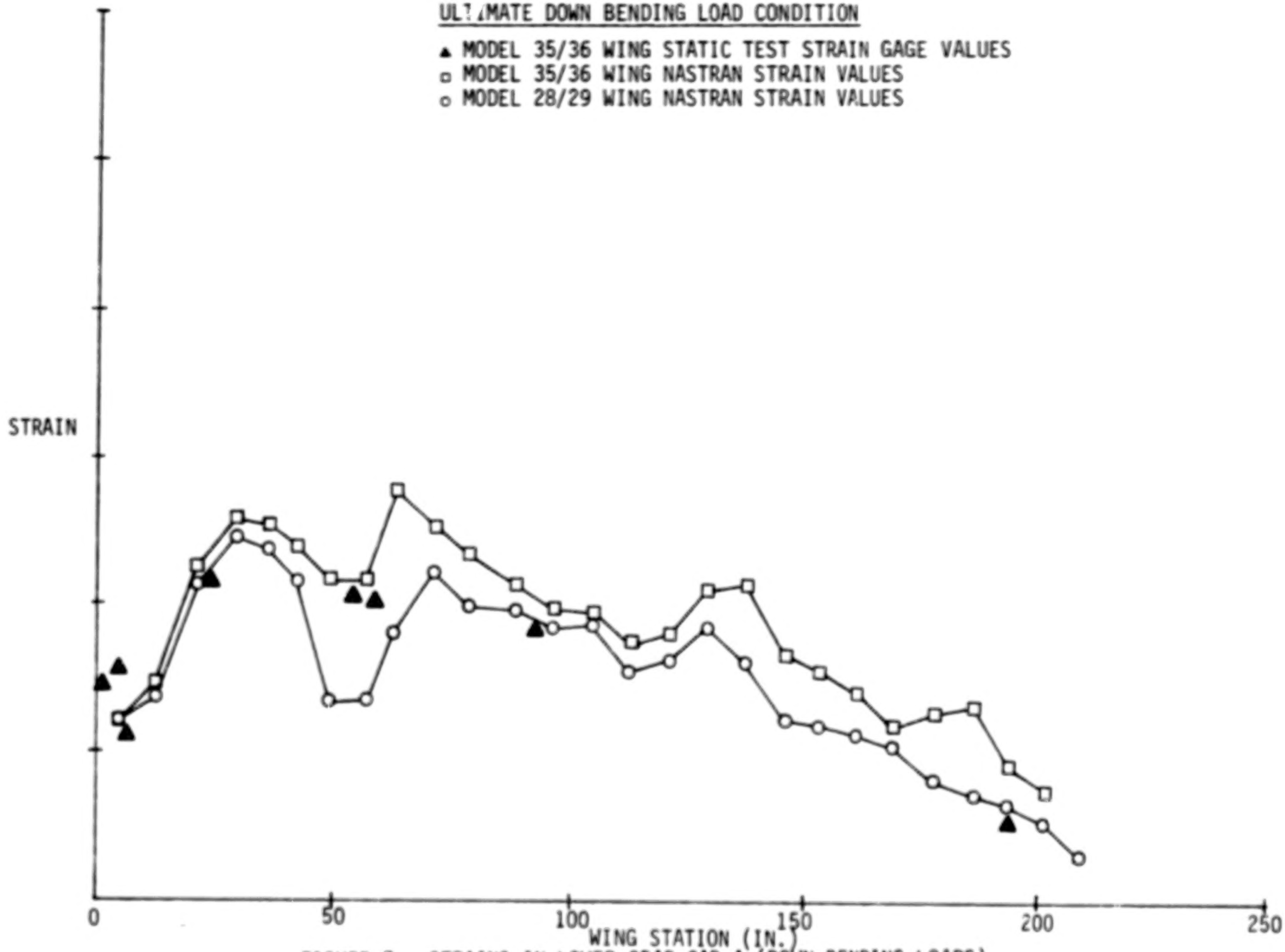


FIGURE 7 - STRAINS IN LOWER SPAR CAP 4 (DOWN BENDING LOADS)

BLANK PAGE

BLANK PAGE

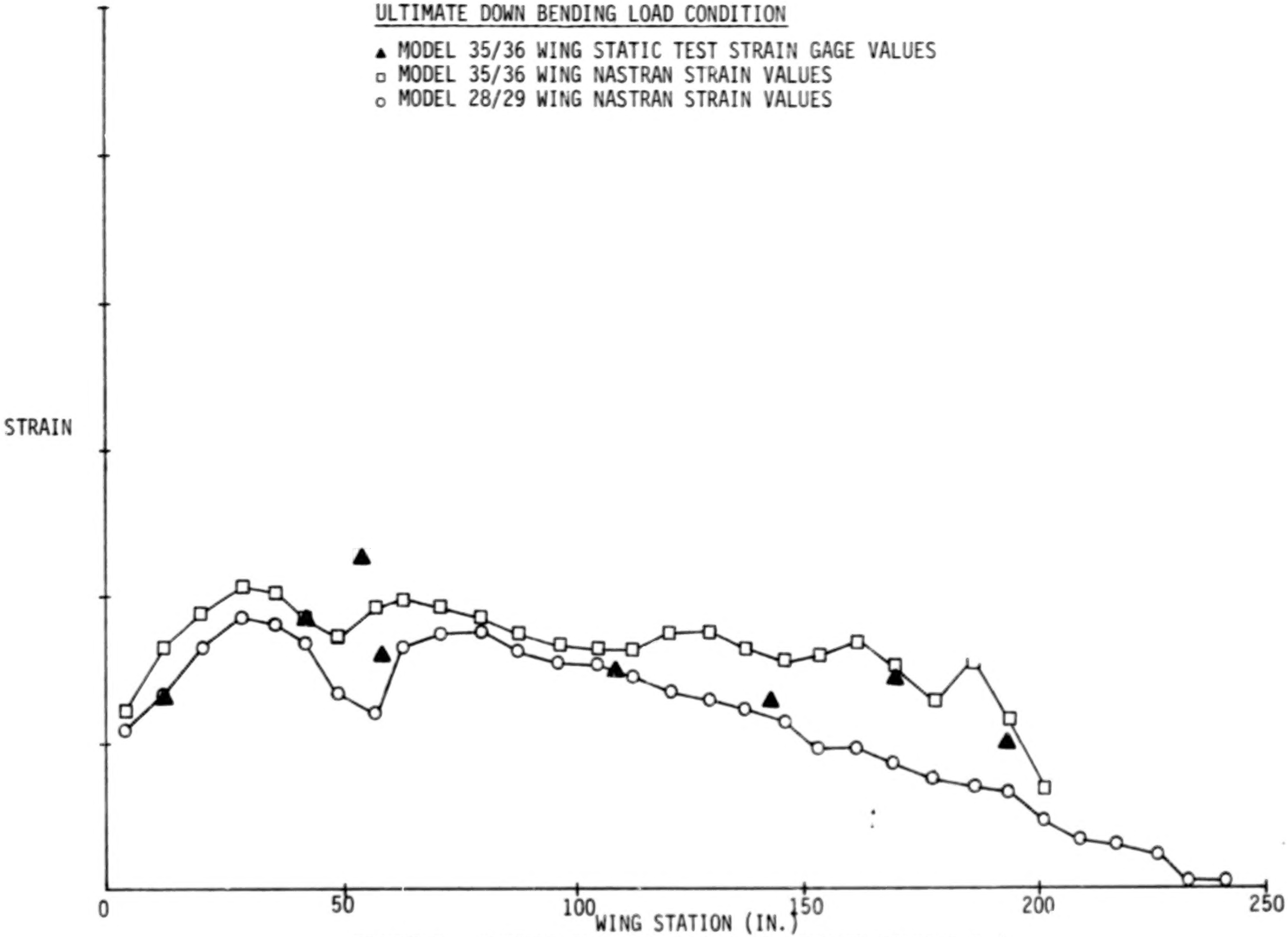


FIGURE 8 - STRAINS IN UPPER WING SKIN BETWEEN SPARS 3 & 4

(DOWN BENDING LOADS)

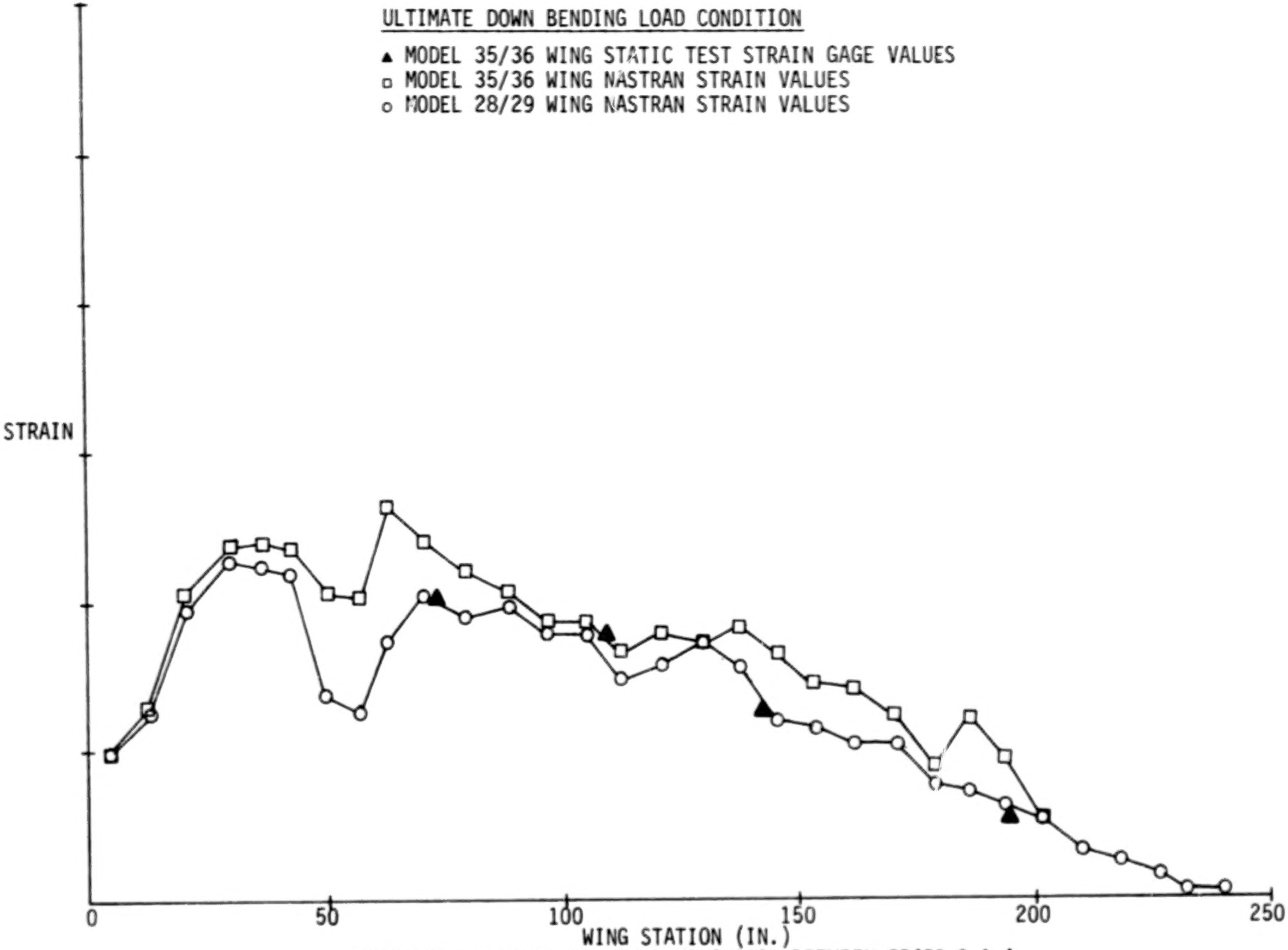


FIGURE 9 - STRAINS IN LOWER WING SKIN BETWEEN SPARS 3 & 4

(DOWN BENDING LOADS)

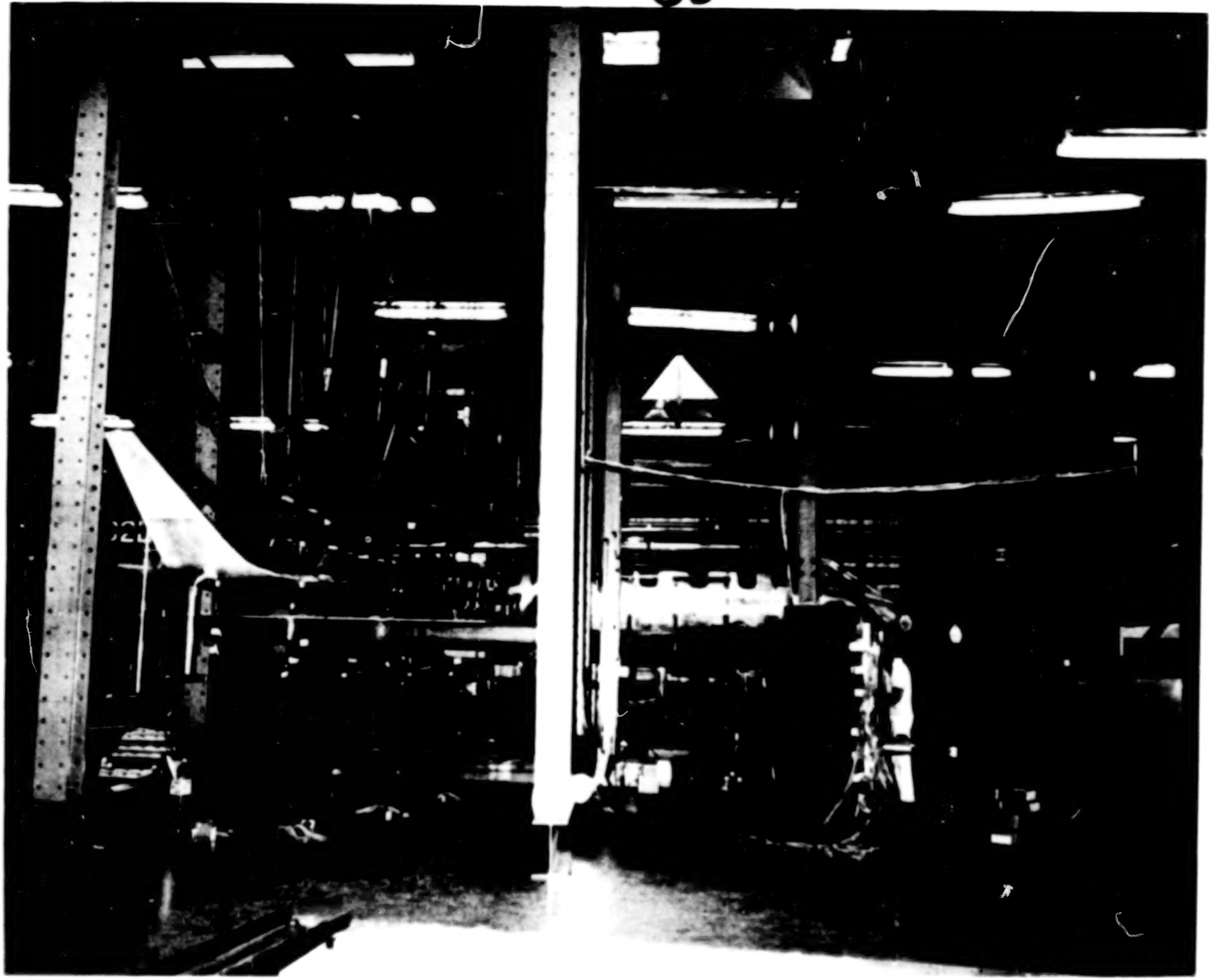


FIGURE 10 - MODEL 28/29 WING STATIC TEST SET-UP

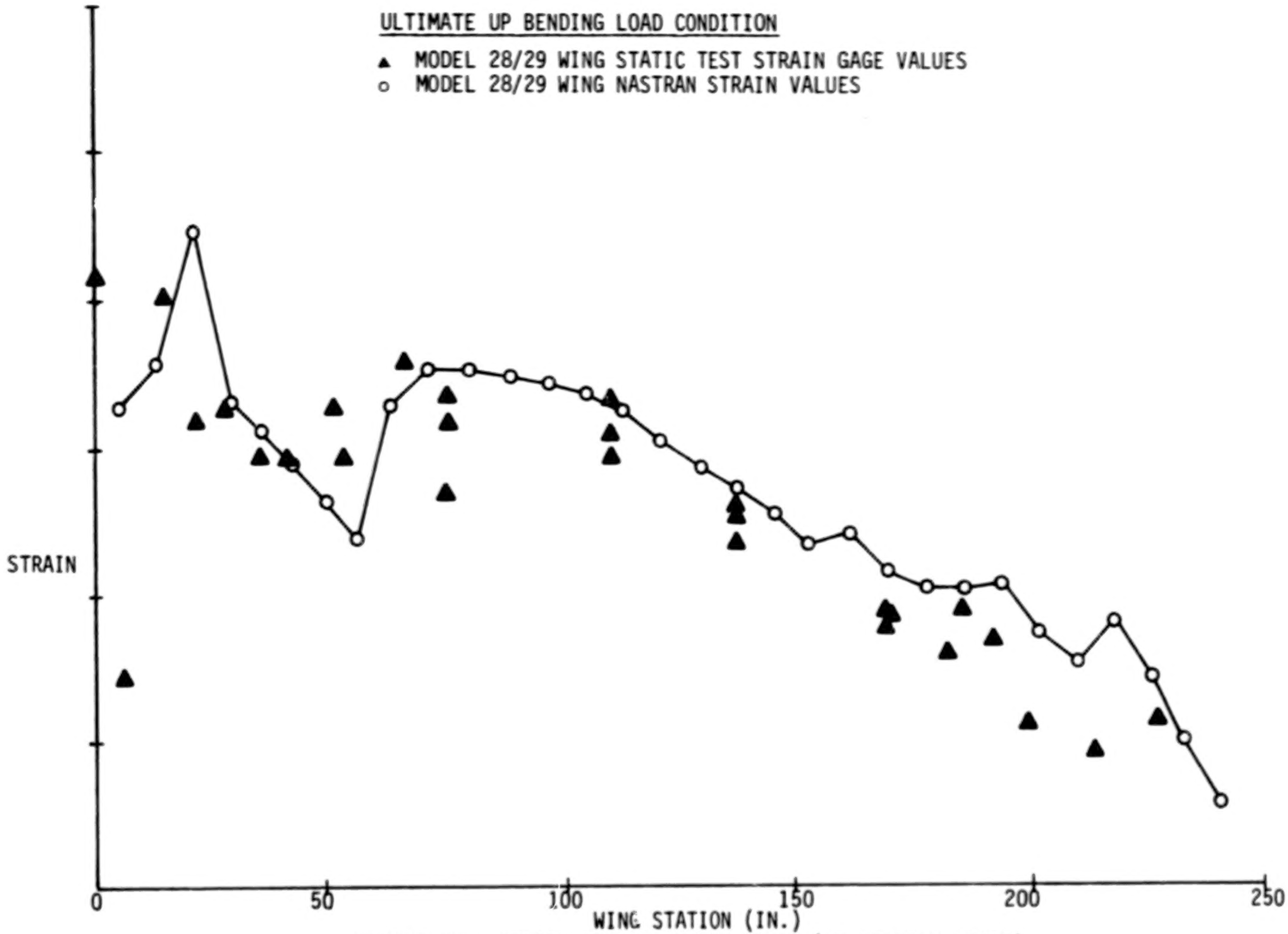


FIGURE 11 - STRAIN IN UPPER SPAR CAP 5 (UP BENDING LOADS)

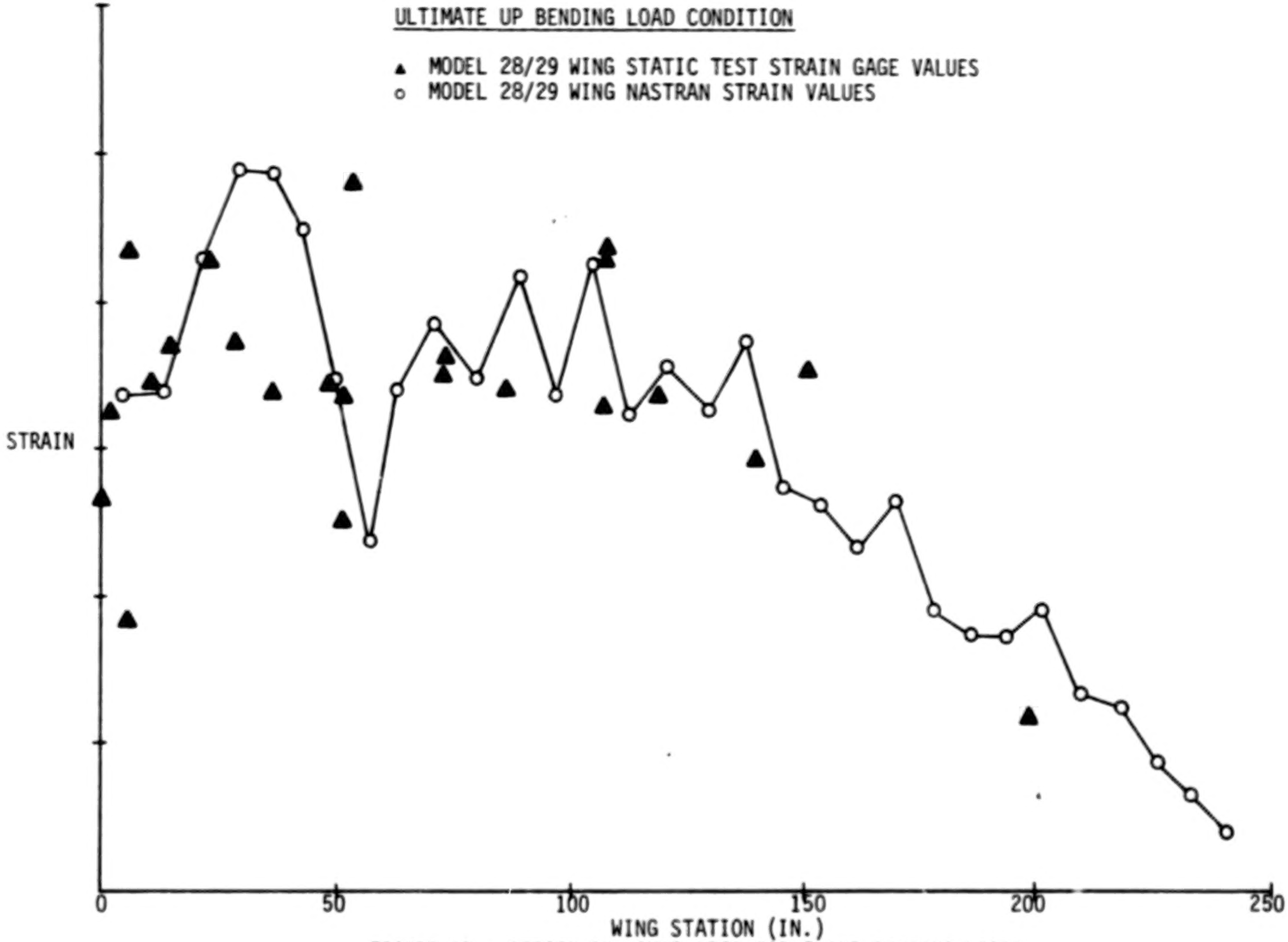


FIGURE 12 - STRAIN IN LOWER SPAR CAP 5 CUP BENDING LOADS

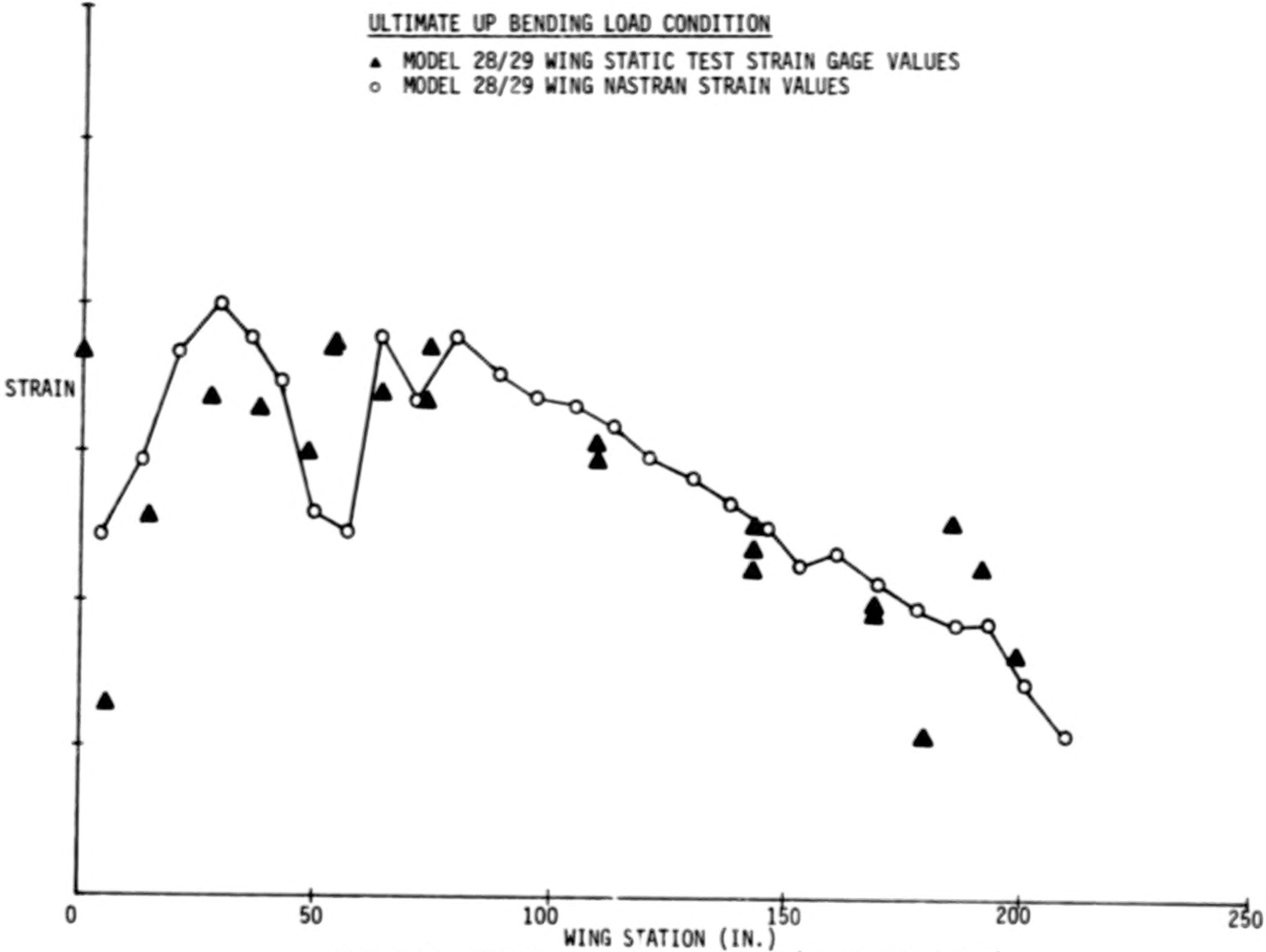


FIGURE 13 - STRAIN IN UPPER SPAR CAP 4 (UP BENDING LOADS)

32

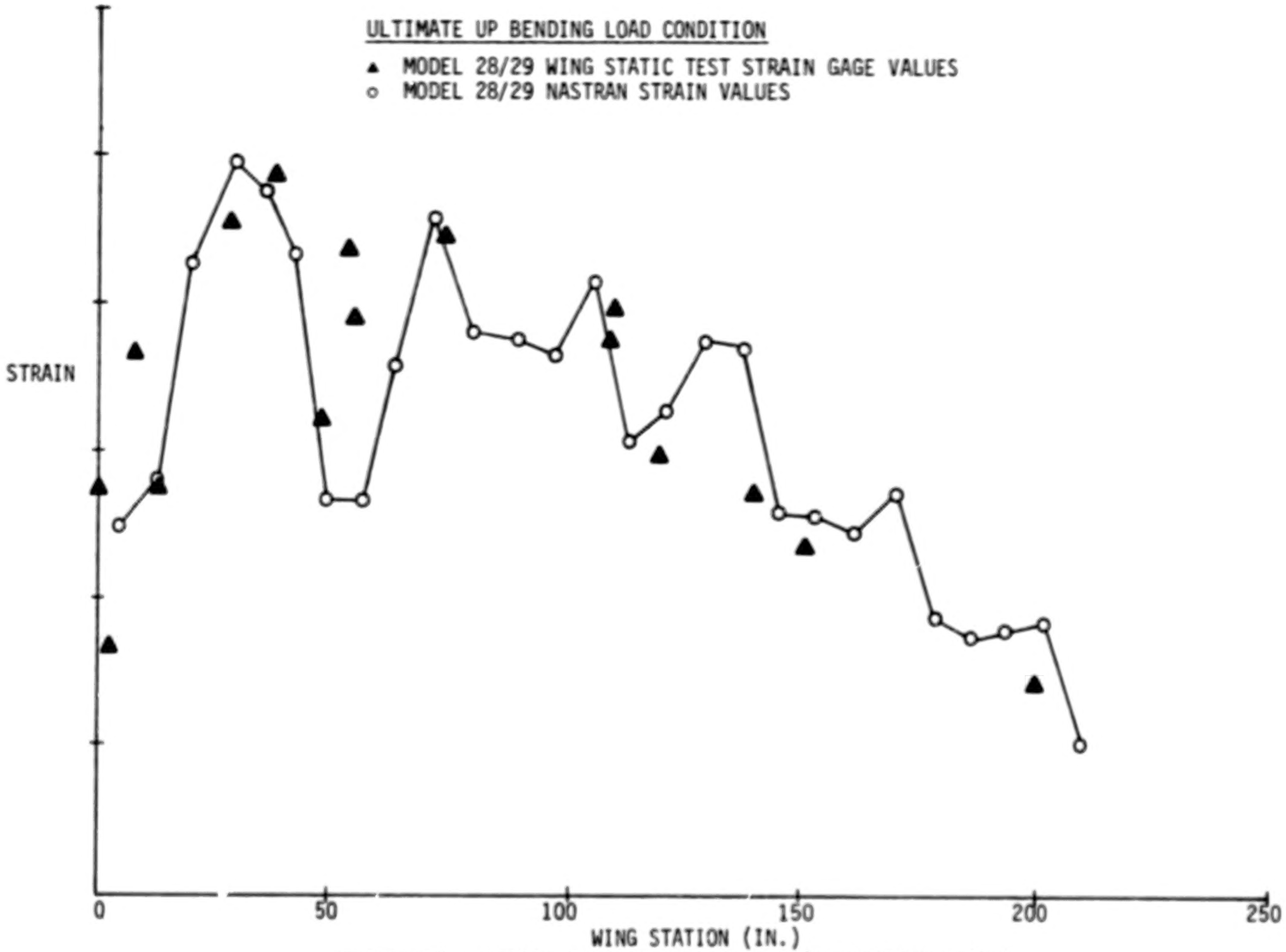


FIGURE 14 - STRAIN IN LOWER SPAR CAP 4 (UP BENDING LOADS)

TENSIONING OF A BELT AROUND A DRUM USING MEMBRANE ELEMENT*

C. H. S. Chen
The BFGoodrich Company

SUMMARY

A class of problem which can be solved by using the membrane element approach is that in which the membrane surface is either unchanged or allowed to undergo a given amount of transverse displacement. A problem belonging to the latter case is the tensioning of a belt around the drum. In this paper the belt tension increase due to drum edge wear or material trapped between the drum and the belt is investigated and some interesting results are obtained. In both cases the increase in belt tension is due to the additional stretching of the belt resulting from the drum radius change rather than from the transverse deflection of the belt.

INTRODUCTION

One shortcoming of the NASTRAN membrane elements is that in their formulations the coupling between bending and stretching is neglected (Ref. 1). In other words, the in-plane strains are independent of the transverse displacements. As a result, the use of NASTRAN membrane elements is very restricted. One class of problems which permits the use of the membrane elements is that in which the membrane surface is either unchanged (such as in a plane stress elasticity problem) or is allowed to undergo given amounts of transverse displacement. In either case, the applied load vectors should always be in the plane of the membrane element. This means that in this class of problems the membrane elements can not take either a normal pressure load or a concentrated load. The problem of belt tensioning around a drum can be classified into that particular class of problem and is used here as an example.

ANALYSIS

This paper demonstrates an application of the membrane element to the problem of the tensioning of a conveyer belt which wraps around a drum. The conveyer belt to be considered has one row of longitudinal wire cable reinforcement placed in a thin sheet of rubber. Conventionally the wire cables are equally spaced across the width of the belt. The dimensions of the belt cross-section and the drum diameter are such that the assumption of the

---

* The author wishes to acknowledge The BFGoodrich Company for the permission to publish this paper.

MICROFILMED FROM
BEST AVAILABLE COPY

belt as a membrane structure in the conventional engineering sense can be justified. The belt is assumed to be orthotropic linear elastic and its in-plane material properties are derived using composite material theory. In particular the well known Halpin-Tsai equations are used by knowing the constituent properties and the wire volume ratio (Ref. 2). The tensioning of the belt considered here comes from the counterweight which is needed to develop sufficient friction between the belt and drum so that the belt can be driven when the driving pulley turns.

We are particularly interested in investigating the stresses developed in the belt under the following situations:

(1) When the drum edges wear out,

(2) When there is material trapped between the belt and the drum,

(3) When both the drum edge wearing and the material trapping occur.

In all three cases there are dangers of an increase in cable tension which affects the belt safety.

The trapped material considered here is in the form of a narrow strip wrapping around the drum for the entire region of belt-drum contact. Figure 1 shows the modeling of a belt using isoparametric quadralateral membrane elements. This belt has 180 degree contact with the drum (called 180 degree wrap angle), but because of symmetry only one half of the wrap angle is included in the model. The drum is not modeled in the problem since it is considered to be rigid. The belt-drum contact region extends from grid point 1 to grid point 7 as shown in Figure 1. The fine grids are used at the edge regions of the belt for convenience in examining different degree of the drum edge wear. The width of the edge regions is arbitrarily assumed.

The tensioning of the belt is simulated by imposing at the straight end of the belt (marked by grid points 11 and 187) a specified amount of uniform displacement while retaning the wrapped end at the half-wrap angle (marked by grid points 1 and 177) unmoved. The drum edge wearing is simulated by allowing the grid points in the belt edge regions to be unrestrained in the radial direction. The trapped materials between the belt and the drum is simulated by specifying at the appropriate grid points within the belt-drum contact a specified amount of radial displacement. For all three cases just mentioned, the loaded belt can be said to have well defined surfaces and thus the membrane elements can be satisfactorily applied.

NUMERICAL RESULTS

The belt under consideration has the following dimensional and material property characteristics:

Ratio of edge region width ($2d_e$) to center region width ($2d_c$) = $2d_e/2d_c$ = 0.35

Ratio of belt thickness (t) to drum radius (R) = t/R = 1/22.529

Ratio of belt thickness (t) to belt total width ($2d_e + 2d_c$) = $t/2(d_e + 2d_c)$ = 1/78.488

Ratio of longitudinal Young's modulus (E_1) to transverse Young's modulus (E_2) = E_1/E_2 = 160.417

Ratio of the height of trapped material (h) to drum radius (R) = h/R =1/31.

The specific amount of displacement required at the straight end (or far end) of the belt to produce a given amount of belt tension is obtained by interpolation between several solutions. Once the element stress (σ_e) is known the cable force (F) can be calculated thus,

$$F = \sigma_e EA/E_1$$

where E and A are the Young's modulus and the cross sectional area of the reinforcing cables respectively.

Figure 2 shows the results of the cable force when there is no drum edge wear, whereas Figure 3 shows that when there is complete drum edge wear. They are all normalized with respect to the ideal uniform cable force (F_o) when the drum has no edge wear and there is no trapped material. F_o is obtained either by dividing the total belt tension by the number of cables or using the equation after the NASTRAN run. The force profiles of Figures 2 and 3 are those along the line of the half-wrap angle where the peaks are always found to be maximum. Five locations of the trapped material are examined. These locations are the belt centerline and four other locations at distances of $d_c/4$, $d_c/2$, $3d_c/4$ and d_c away from the belt centerline. Here d_c is the half-width of the belt center region.

It can be seen from Figures 2 and 3 that, for this particular case, the cable force (or stress) increases by about 3.3 times due to material trapping alone, about 1.3 times due to drum edge wear alone and about 4 times due to the combination of the two. These clearly indicate the danger of material trapping and excessive drum edge wear. Figures 2 and 3 also reveal that the force or stress concentration is highly local in nature and that the peak values appear to be independent of the location of the trapped material. These imply that the interactions between the cables are not very strong and they decay fast. This fact suggests that in the simplified analysis the cable interaction can be neglected.

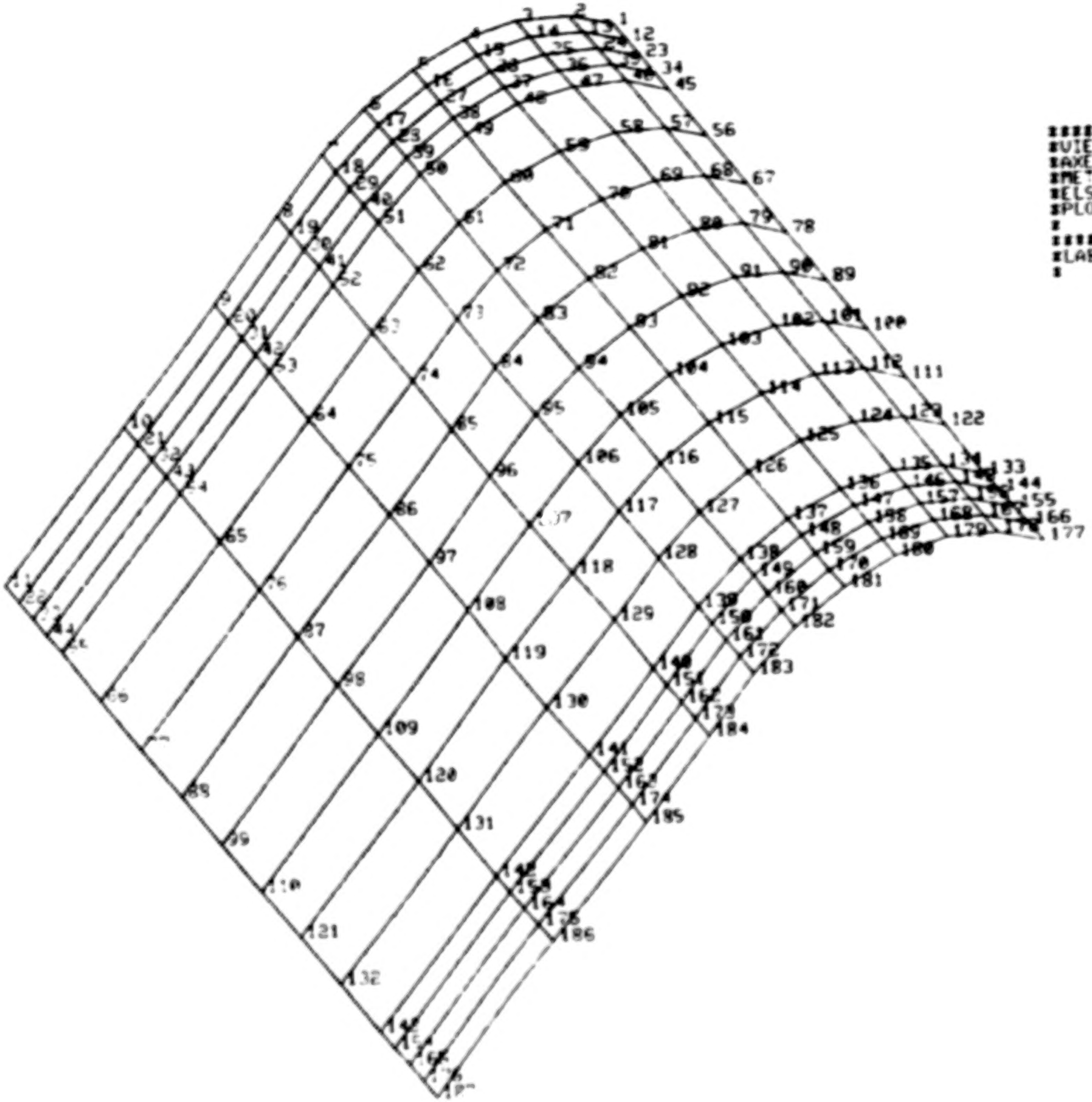
CONCLUDING REMARKS

Using the problem of tensioning of a belt as an example, the present paper has pointed out a class of problems which can be adequately solved by using NASTRAN membrane elements. This class of problems are such that the middle surface of the membrane is either undeformed or undergoes a specified amount of deflection. We have studied the effects of material trapping and drum edge wear on the cable tension in a belt. It should be noted here that the material trapped between the drum and belt is of a special form that has the equvalent effect of increasing the drum radius. The increase of the belt tension really comes from the additional stretching of the belt resulting from the growth of the drum radius rather than due to the transverse deflection of the membrane.

REFERENCES

1. The NASTRAN Theoretical Manul, NASA SP-221(04), Dec. 1977.

2. Jones,R.M., Mechanics of Composite Materials, McGraw-Hill Book Co., 1975.



1
2
3
12
14
23
34
45
47
49
50
51
52
53
54
56
57
58
59
60
61
62
64
65
67
68
69
70
71
72
73
74
75
76
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
135
136
137
138
139
140
144
146
147
148
149
150
155
157
158
159
160
161
166
168
169
170
171
172
177
179
180
181
182
183
184
185
186

```
*****
*VIEW 45.,30.,45.
*AXES X,Y,Z
*METHOD 2
*ELSET ALL
*PLOT
*
*****
*LABEL
*
```

FIGURE 1 - BELT MODEL

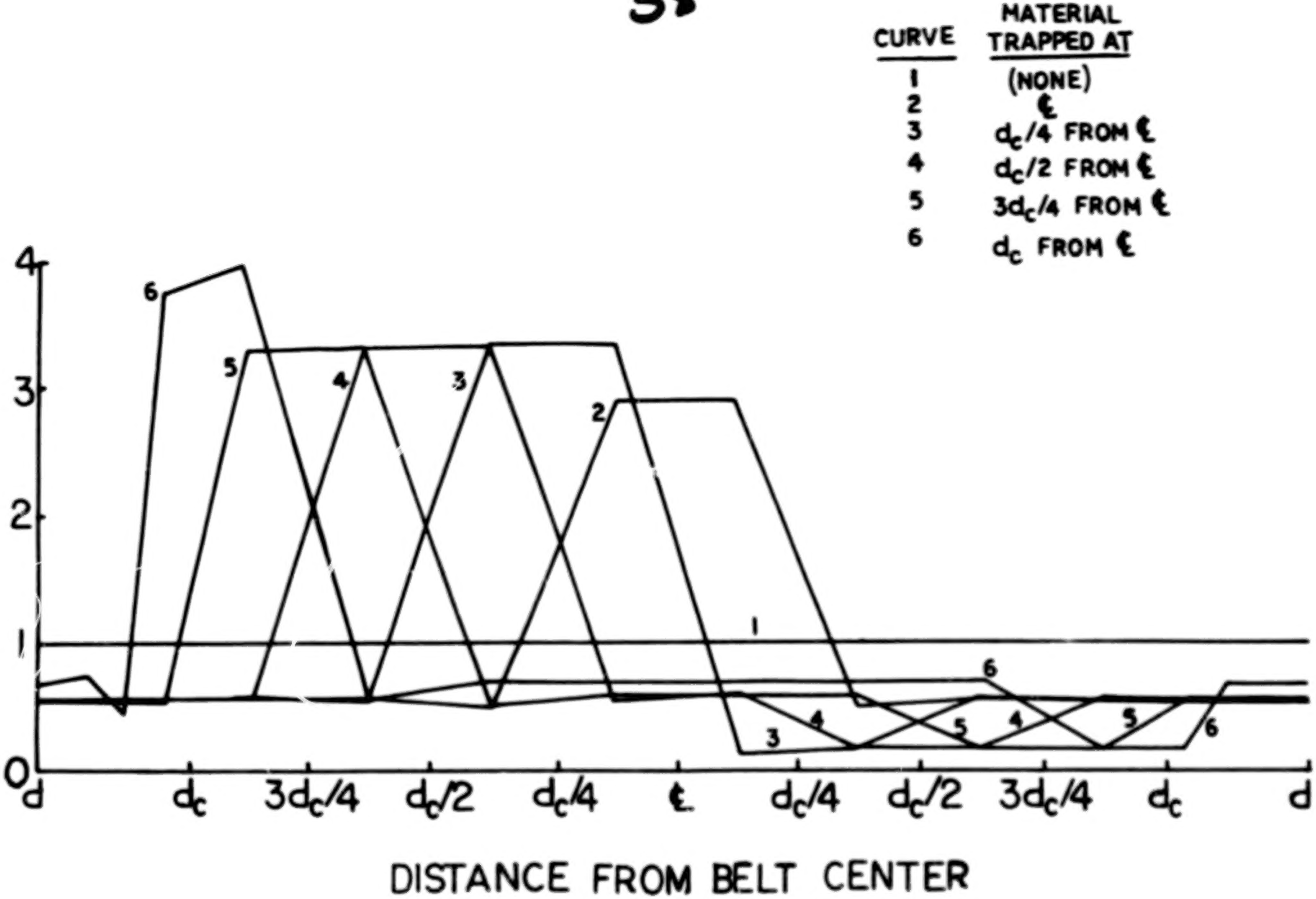


FIGURE 2 - CABLE FORCE PROFILE - NO EDGE WEAR OF DRUM

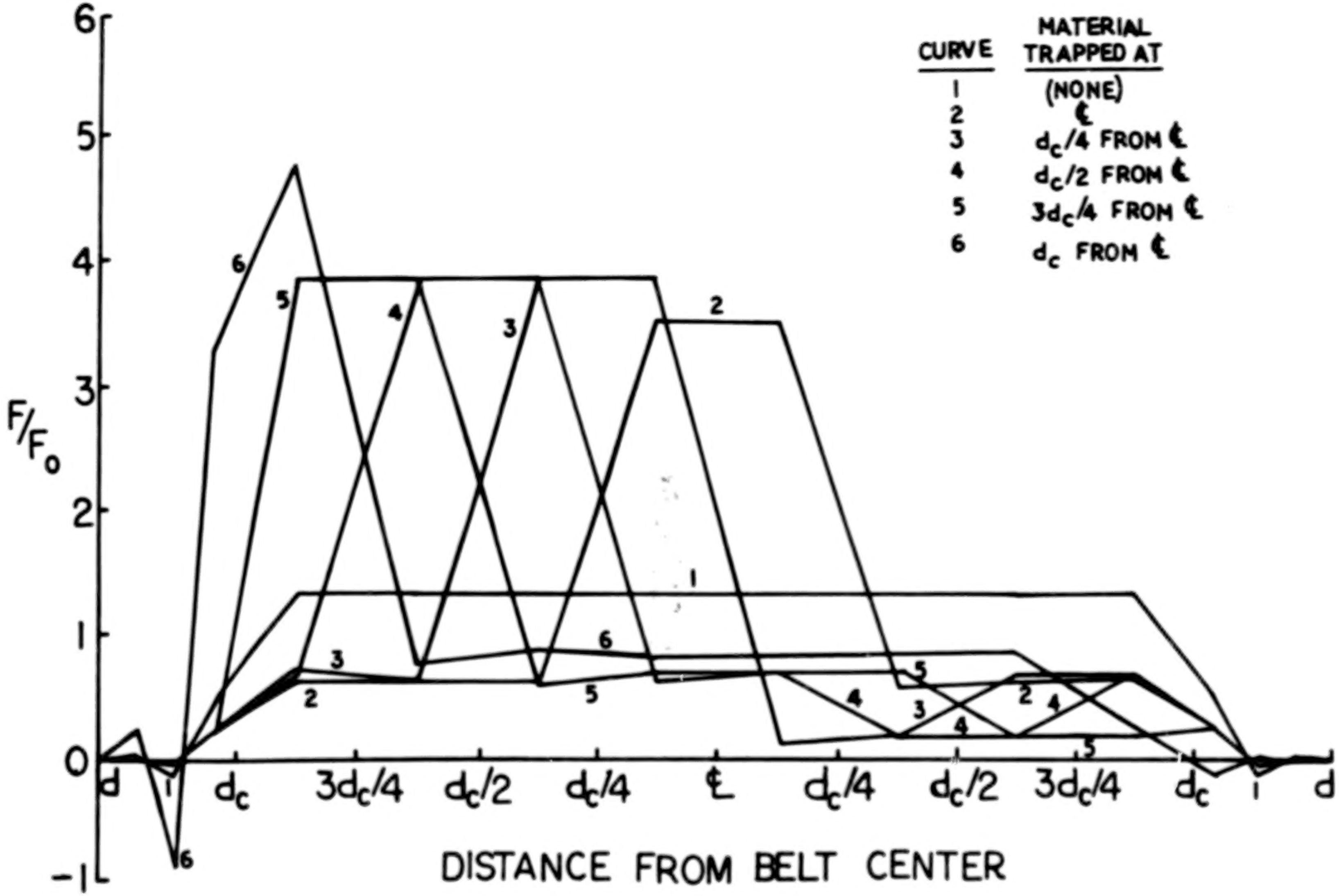


FIGURE 3 - CABLE FORCE PROFILE - COMPLETE EDGE WEAR OF DRUM

BLANK PAGE

THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS SUBJECTED TO ASYMMETRIC THERMAL LOADS*

Gordon R. Negaard
ANAMET Laboratories, Incorporated

SUMMARY

This paper presents a study of the performance of the NASTRAN level 16.0 axisymmetric solid elements when subjected to both symmetric and asymmetric thermal loading. A ceramic radome was modeled using both the CTRAPRG and the CTRAPAX elements. The thermal loading applied contained severe gradients through the thickness of the shell. Both elements were found to be more sensitive to the effect of the thermal gradient than to the aspect ratio of the elements. Analysis using the CTRAPAX element predicted much higher thermal stresses than the analysis using the CTRAPRG element, prompting studies of models for which theoretical solutions could be calculated. It was found that the CTRAPRG element solutions were satisfactory, but that the CTRAPAX element was very geometry dependent. This element produced erroneous results if the geometry was allowed to vary from a rectangular cross-section. The most satisfactory solution found for this type of problem was to model a small segment of a symmetric structure with isoparametric solid elements and apply the cyclic symmetry option in NASTRAN.

INTRODUCTION

Two recent studies have been conducted to determine stresses in ceramic radomes due to asymmetric thermal loadings. Transient thermal loads in both studies produced much sharper temperature gradients through the thickness of the shell than along the surface. For this reason, shell elements could not be used and it was necessary to use a formulation capable of modeling a three-dimensional temperature distribution. In the first study, four layers of NASTRAN level 16.0 twenty node isoparametric bricks (CIHEX2) of various thicknesses were used to construct a radome model (Fig. 1). The thermal loading simulated a threat level laser irradiation. The results were found to be very dependent upon matching the nodal spacing to the temperature distribution and a problem size limitation was reached where economics prohibited creating a finer model which would be less sensitive to the temperature gradient.

In the second study, the thermal loading simulated both axisymmetric and non-axisymmetric aerodynamic heating. The structure was modeled with a CTRAPAX axisymmetric element capable of handling both loading cases. The nodal point temperatures for an axisymmetric thermal load case are shown in Figure 2, which also illustrates the modeling of the radome nose-tip. The grid illustrated represents the third iteration of the mesh size. The initial grid had the same spacing through the thickness but was several times coarser in the axial direction. This resulted in temperature gradients between nodal points of as much as 600°R. The NASTRAN results predicted unrealistically high stresses which were at first thought to be a function of the temperature gradient. The grid shown in Figure 2 reduced these gradients to less than 300°R, however, the stress levels were still not believable. A switch to CTRAPRG elements produced maximum compressive stresses of about 21000 psi, which agreed well with a SAAS III analysis and indicated that the CTRAPAX elements were indeed predicting erroneous stresses. These cases had been conducted with temperature dependent material properties so both elements were run with the temperature dependence removed, however this changed the results only slightly, eliminating this also as a possible source of the error.

---

*The work described herein was performed by the Aerospace Structures Information & Analysis Center (ASIAC) at the Air Force Flight Dynamics Laboratory, WPAFB, Ohio under Air Force Contract F33615-77-C-3046.

At this point, the reason for the variations in the stresses predicted by the two elements were unknown. A preliminary study on a hollow cylinder had shown almost identical answers for both a linearly and logarithmically varying radial temperature distribution for the two elements. Various ways of modeling the radome with triangles and quadrilateral elements were investigated to determine if the problem was a function of modeling techniques. This did not appear to be so since all combinations of the AX elements produced similar stresses and the RG elements likewise produced a set of similar stresses. Figures 3 and 4 compare hoop stresses for the two elements and show that the CTRAPAX element predicts a ridiculously high stress of more then 600,000 psi in the same area. Since only the CTRAPAX and CTRIAAX elements could handle asymmetric loading, it was necessary to determine the reliability of these elements before continuing with the asymmetric aerodynamic heating case.

SYMBOLS

α = thermal expansion coefficient

ν = poisson's ratio

E = young's modulus

T = temperature field

σ = stress

t = thickness

b = radius

DISC ANALYSIS

In order to determine why the CTRAPAX and CTRAPRG elements produced differing results for the radome model, a simple disc model restrained at the outer circumference and which had known theoretical solutions was chosen (ref. 1). Both a quadratically varying radial temperature and a linearly varying axial temperature could be applied as shown in Table 1. The model and cross-sections of the axisymmetric model using rectangular elements are shown in Figure 7. For this model, both types of elements produced exact theoretical answers for aspect ratios varying from 1.0 to 10.0 for temperature independent material properties for combinations of radially and axially varying temperatures. This eliminated aspect ratios and two-dimensional temperature gradients as being responsible for the differing results in the radome. There was no theoretical solution for temperature dependent material properties, however, both elements still produced identical answers. This left geometry of the elements as the only likely remaining candidate for the source of error.

The geometry of the elements was changed so that all elements had at least one skewed side instead of being rectangular (Fig. 7). This model was then run with only the radially varying temperature distribution. These runs produced results that definitely proved that the CTRAPAX elements produced incorrect results when a non-rectangular cross-section is used. For the temperature independent results, the CTRAPRG elements produced results which matched the theoretical solution exactly, while the CTRAPAX elements gave radial and hoop stresses sixty to one hundred per cent too high. Even worse, these elements predicted axial stresses almost as high as the axial and radial stresses while the CTRAPRG results agreed with the theoretical solution of zero stress. The temperature dependent material runs predicted stresses that followed the same pattern but of course could not be compared to a theoretical solution. These results are shown in Figures 8 through 11.

This analysis showed that the CTRAPRG element appeared to produce reliable results while confirming that the CTRAPAX element could not be trusted in a model requiring the use of non-rectangular element shapes, essentially ruling out the use of the CTRAPAX element in a model simulating a radome shape. Since only the CTRAPAX and CTRIAAX elements can be used for axisymmetric models subjected to non-axisymmetric loads, it was necessary to look for an alternate way of solving the asymmetric aerodynamic heating problem.

RING ANALYSIS

The cyclic symmetry option in NASTRAN was examined to determine if better results for symmetric structures subjected to an asymmetric thermal load could be obtained. A ring subjected to a temperature distribution of the form $T=T_0 (R^K) \cos(n\theta)$ was chosen because theoretical solutions could be obtained (ref.2). The model of the ring is shown in Figure 12. Both thirty degree and ten degree wedge shapes were examined, requiring twelve and thirty-six cases respectively when running cyclic symmetry. The model cross-section was deliberately made as similar to the previous disc analysis as possible including the use of skewed elements shapes exactly as used in the previous section. Table 2 describes the formulation of the temperatures used as input and the resulting hoop and radial stresses to be expected. The axial stress should be identically zero. The actual stresses obtained at several radii are shown in Tables 3 and 4 for both ten and thirty degree wedges with skewed and non-skewed geometry. It can be seen that skewness had little effect except for the radial stress in the outermost elements in the ten degree wedge. The axial stresses tended to be less than ten percent of the lower of the radial or hoop stress except at the outer fiber. It was discovered that the axial stresses could be made smaller by making the ring thinner, thus approaching a state of plane stress more closely. Selected plots of hoop stress are shown as Figures 13, 14 and 15. It can be seen in these figures that as the wedge becomes narrower, it appears to approach the theoretical solution as a limit.

CONCLUDING REMARKS

In order to make a comparison of the computer costs of cyclic symmetry against the use of the axisymmetric elements, a ring model with the same geometry using the CTRAPAX elements with non-skewed geometry and temperature input at every fifteen degrees as shown in Table 1 was examined. This produced almost exact theoretical answers, remembering that the CTRAPAX element required that the element shapes be rectangular while the cyclic symmetry option did not have this limitation. The following is a comparison of the running time on the CDC CYBER 74, using level 16.0 NASTRAN with 32 elements in each model.

	CYCLIC SYMMETRY TECHNIQUE		AXISYMMETRIC TECHNIQUE
	(30° Wedge)	(10° Wedge)	(15° Increments)
CM(octal)	165,000	170,000	250,000
CP(sec)	445	1,200	2,681
IO(sec)	333	869	307

These results indicate that the cyclic symmetry option in NASTRAN is better suited to the solution of a general axisymmetric problem under asymmetric loading than the axisymmetric technique. A practical upper limit to the size problem that can be solved with cyclic symmetry remains to be determined.

REFERENCES

1. Wang, C. T., "Applied Elasticity " McGraw-Hill Book Company, Inc., 1953.

2. Maddux, G. E., "Thermo-Structural Analysis Manual," Report No. WADD-TR-60-517, Vol. 1, August, 1962.

TABLE 1. PARAMETERS USED IN DISC MODEL

DISC PROPERTIES

b (OUTER RADIUS = 10.0 INCHES

t (THICKNESS) MAX = 0.4 INCH

t (THICKNESS) MIN = 0.04 INCH

$E = 10^7$ PSI

$\nu = 1/3$

$\alpha = 10^{-6}$ IN/IN/°R

RADIAL TEMPERATURE AND STRESS VARIATION

$$T(r) = T_o + (T_i - T_o)\left[1 - \frac{r^2}{b^2}\right]$$

$$\sigma_{rr}(r) = -\tfrac{1}{4} E\alpha (T_i - T_o)\left[\frac{3-\nu}{1-\nu} - \frac{r^2}{b^2}\right]$$

$$\sigma_{\theta\theta}(r) = -\tfrac{1}{4} E\alpha (T_i - T_o)\left[\frac{3-\nu}{1-\nu} - \frac{3r^2}{b^2}\right]$$

$$\sigma_{zz}(r) = 0$$

AXIAL TEMPERATURE AND STRESS VARIATION

$$T(z) = 1000\,z$$

$$\sigma_{rr}(z) = \sigma_{\theta\theta}(z) = \frac{E\alpha\ T(z)}{2(1-\nu)}$$

$$\sigma_{zz}(z) = 0$$

TABLE 2. PARAMETERS USED IN RING MODEL

RING PROPERTIES

b (OUTER RADIUS) = 1.0 INCH

a (INNER RADIUS) = 0.2 INCH

t (THICKNESS) = 0.08 INCH

$E = 10^7$ PSI

$\upsilon = 1/3$

$\alpha = 10^{-6}$ IN/IN/°R

TEMPERATURE AND STRESS VARIATION

$$T(r,\theta) = T_o \left(\frac{r}{b}\right)^2 \cos\theta$$

$$\sigma_{rr}(r,\theta) = E\,\alpha\,T_o\,B_k \cos\theta$$

$$\sigma_{\theta\theta}(r,\theta) = E\,\alpha\,T_o\,D_k \cos\theta$$

r	B_k	D_k
.2	0.0	.1202
.3	.03258	.1177
.4	.04408	.1163
.5	.04808	.1024
.6	.04696	.07364
.7	.04144	.02929
.8	.03170	−.03088
.9	.01788	−.1070
1.0	0.0	−.1990

TABLE 3. STRESSES IN 30° WEDGE MODEL AS A FUNCTION OF ANGLE AND RADIUS

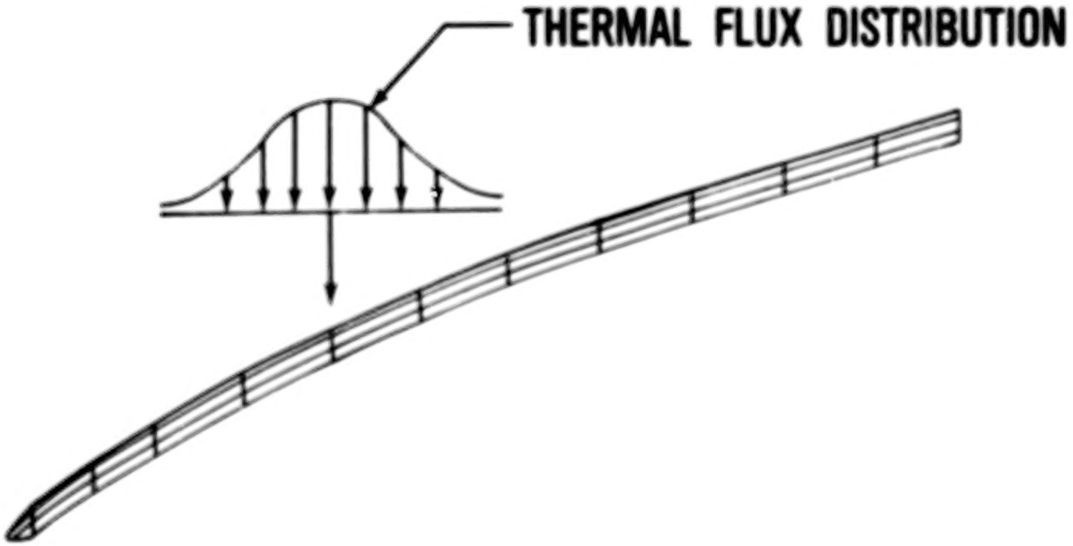
RADIUS (INCHES)	THETA (DEGREES)	RING WITH UNSKEWED ELEMENTS RADIAL	HOOP	AXIAL	RING WITH SKEWED ELEMENTS RADIAL	HOOP	AXIAL
.25	0	41	187	-4	42	187	-3
	30.0	35	162	-4	36	162	-2
	60.0	20	93	-2	21	94	-2
	90.0	0	0	0	0	0	0
.55	0	87	158	8	87	158	9
	30.0	76	137	7	76	137	8
	60.0	43	79	4	44	79	5
	90.0	0	0	0	0	0	0
.95	0	26	-221	43	26	-221	43
	30.0	23	-192	37	22	-192	37
	60.0	13	-111	22	13	-111	21
	90.0	0	0	0	0	0	0

TABLE 4. STRESSES IN 10° WEDGE MODEL AS A FUNCTION OF ANGLE AND RADIUS

		RING WITH UNSKEWED ELEMENTS			RING WITH SKEWED ELEMENTS		
RADIUS (INCHES)	THETA (DEGREES)	RADIAL	HOOP	AXIAL	RADIAL	HOOP	AXIAL
.25	0	46	224	-8	50	224	-5
	30	40	194	-7	43	194	-5
	60	23	112	-4	25	112	-3
	90	0	0	0	0	0	0
.55	0	101	178	0	102	180	2
	30	87	154	0	89	156	1
	60	50	89	0	51	90	1
	90	0	0	0	0	0	0
.95	0	28	-276	33	18	-279	33
	30	25	-239	28	16	-242	29
	60	14	-138	16	9	-140	17
	90	0	0	0	0	0	0

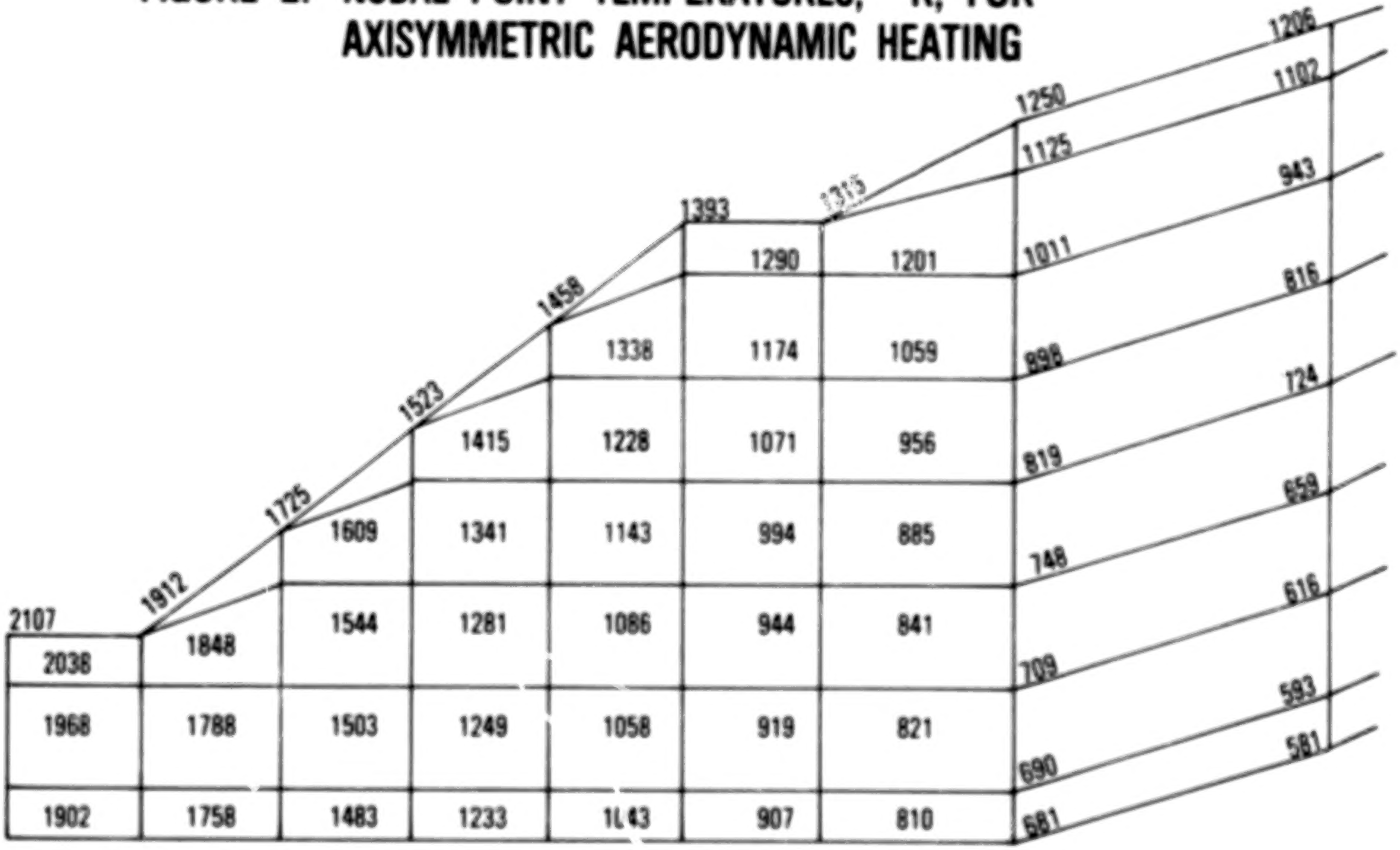
FB 7-9-114 —4

FIGURE 1. TYPICAL AXISYMMETRIC RADOME MODEL WITH ASYMMETRIC HEAT LOAD


THERMAL FLUX DISTRIBUTION

FB 79112-4

FIGURE 2. NODAL POINT TEMPERATURES, °R, FOR AXISYMMETRIC AERODYNAMIC HEATING


1206
1102
1250
1125
943
1393
1315
1290
1201
1011
816
1458
1338
1174
1059
898
724
1523
1415
1228
1071
956
819
659
1725
1609
1341
1143
994
885
748
1912
2107
2038
1848
1544
1281
1086
944
841
616
709
1968
1788
1503
1249
1058
919
821
593
690
1902
1758
1483
1233
1043
907
810
581
681

BLANK PAGE

BLANK PAGE

FIGURE 3. HOOP STRESS (PSI) FOR CTRAPAX ELEMENT

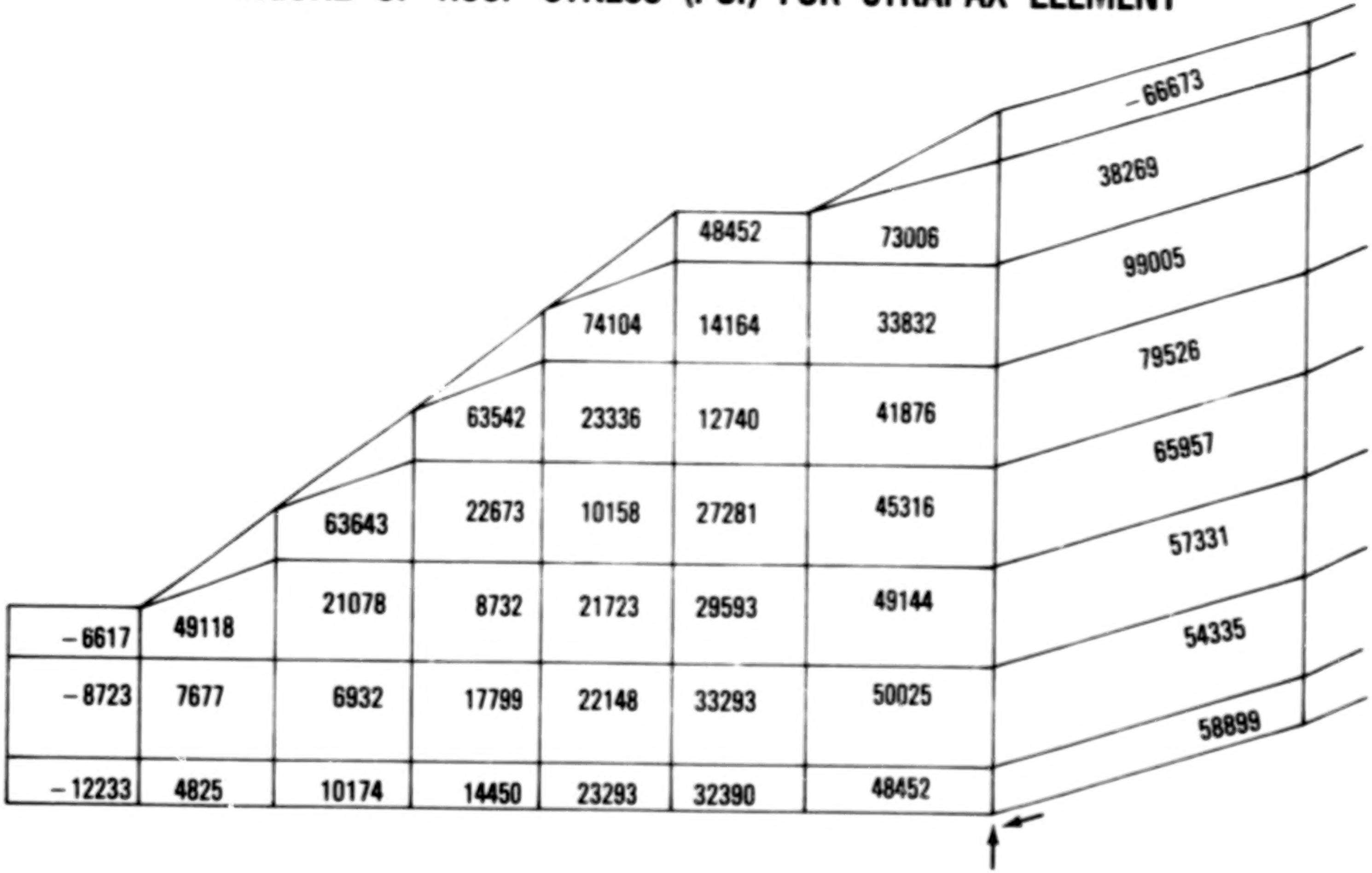
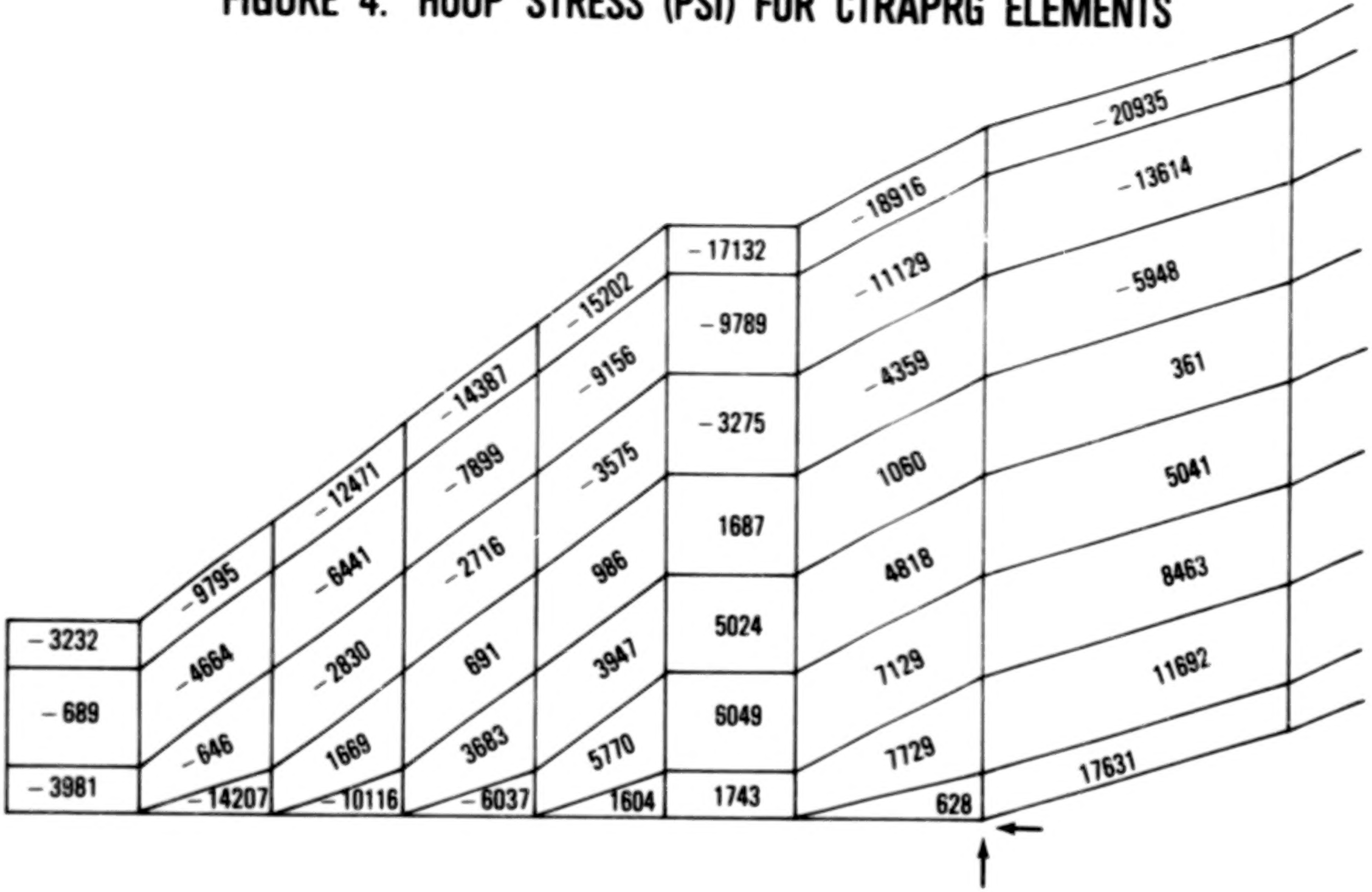
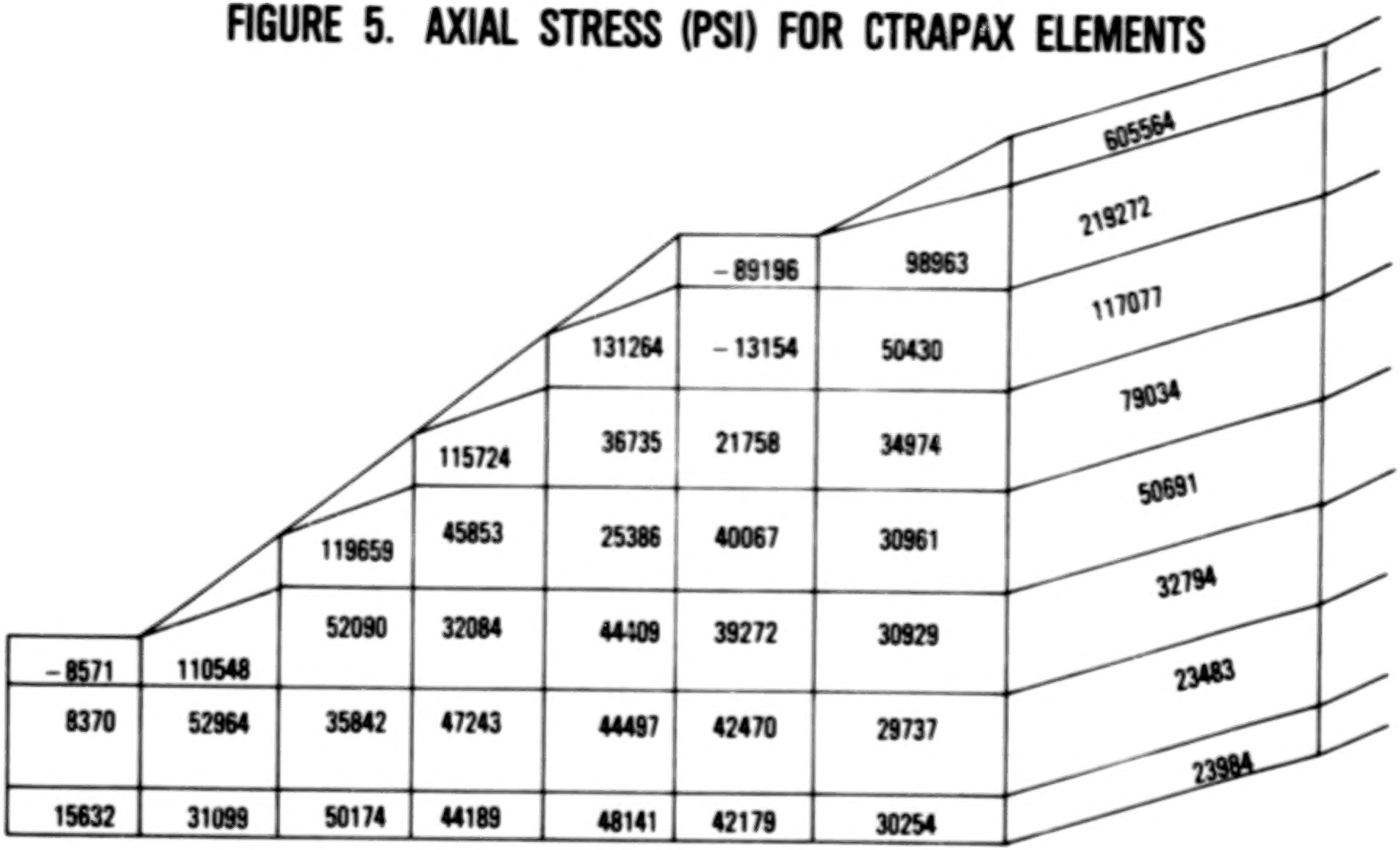

-66673
38269
48452
73006
99005
74104
14164
33832
79526
63542
23336
12740
41876
65957
63643
22673
10158
27281
45316
57331
21078
8732
21723
29593
49144
54335
-6617
49118
-8723
7677
6932
17799
22148
33293
50025
58899
-12233
4825
10174
14450
23293
32390
48452

FIGURE 4. HOOP STRESS (PSI) FOR CTRAPRG ELEMENTS


-3232
-689
-3981
-9795
-4664
-646
-14207
-12471
-6441
-2830
1669
-10116
-14387
-7899
-2716
691
3683
-6037
-15202
-9156
-3575
986
3947
5770
1604
-17132
-9789
-3275
1687
5024
6049
1743
-18916
-11129
-4359
1060
4818
7129
7729
628
-20935
-13614
-5948
361
5041
8463
11692
17631

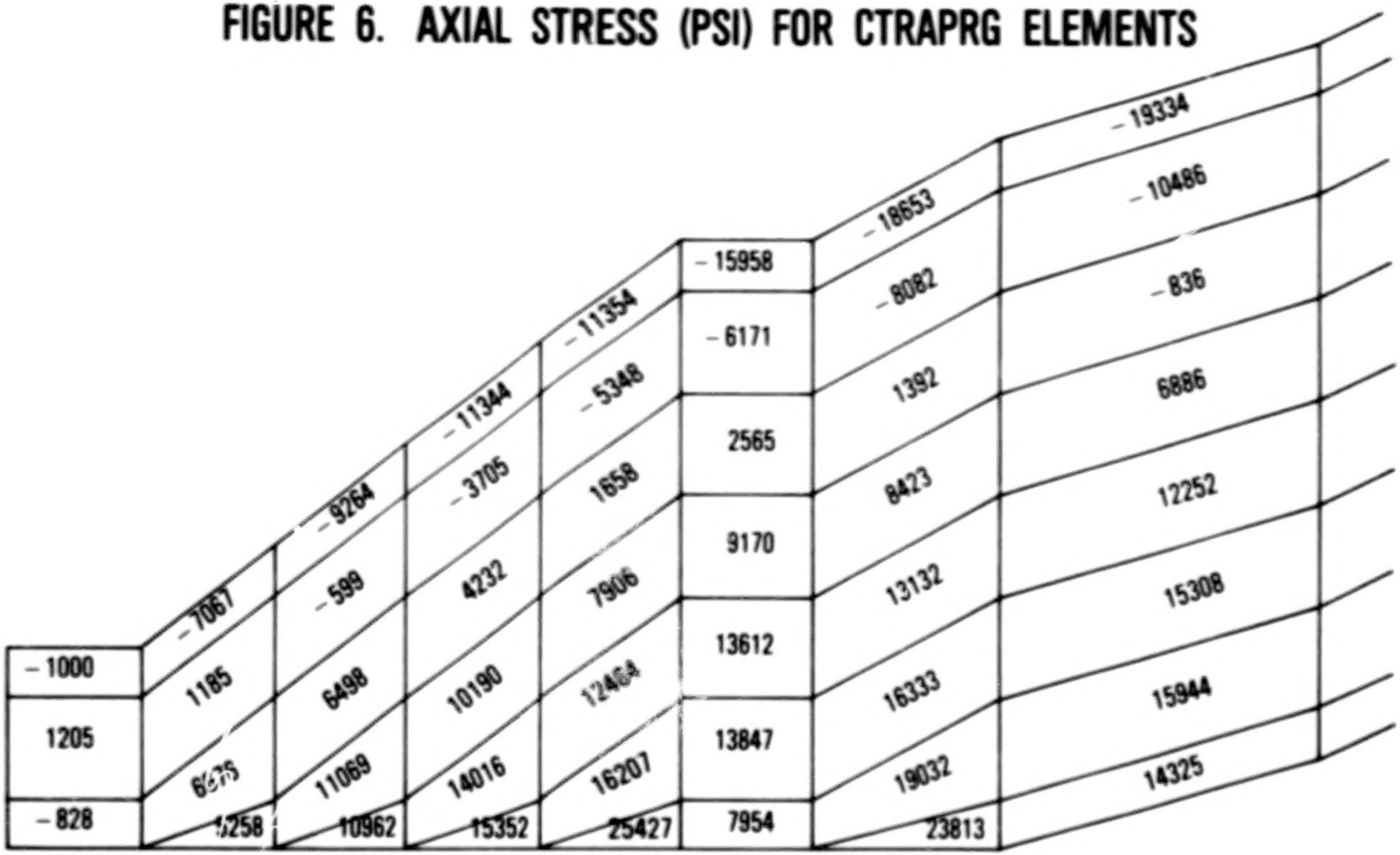
FB -7-9-109 —4

FIGURE 5. AXIAL STRESS (PSI) FOR CTRAPAX ELEMENTS


605564
219272
-89196
98963
131264
-13154
50430
117077
115724
36735
21758
34974
79034
119659
45853
25386
40067
30961
50691
-8571
110548
52090
32084
44409
39272
30929
32794
8370
52964
35842
47243
44497
42470
29737
23483
15632
31099
50174
44189
48141
42179
30254
23984

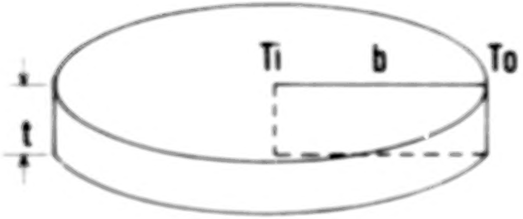
FB-7-9-108—4

FIGURE 6. AXIAL STRESS (PSI) FOR CTRAPRG ELEMENTS



– 1000
1205
– 828
– 7067
1185
6635
5258
– 9264
– 599
6498
11069
10962
– 11344
– 3705
4232
10190
14016
15352
– 11354
– 5348
1658
7906
12464
16207
25427
– 15958
– 6171
2565
9170
13612
13847
7954
– 18653
– 8082
1392
8423
13132
16333
19032
23813
– 19334
– 10486
– 836
6886
12252
15308
15944
14325

FIGURE 7. AXISYMMETRIC DISC MODEL



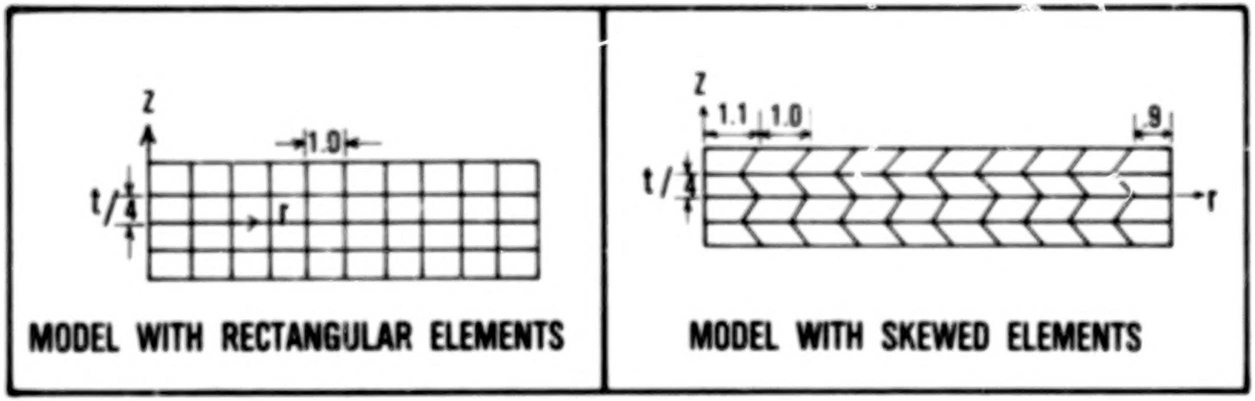
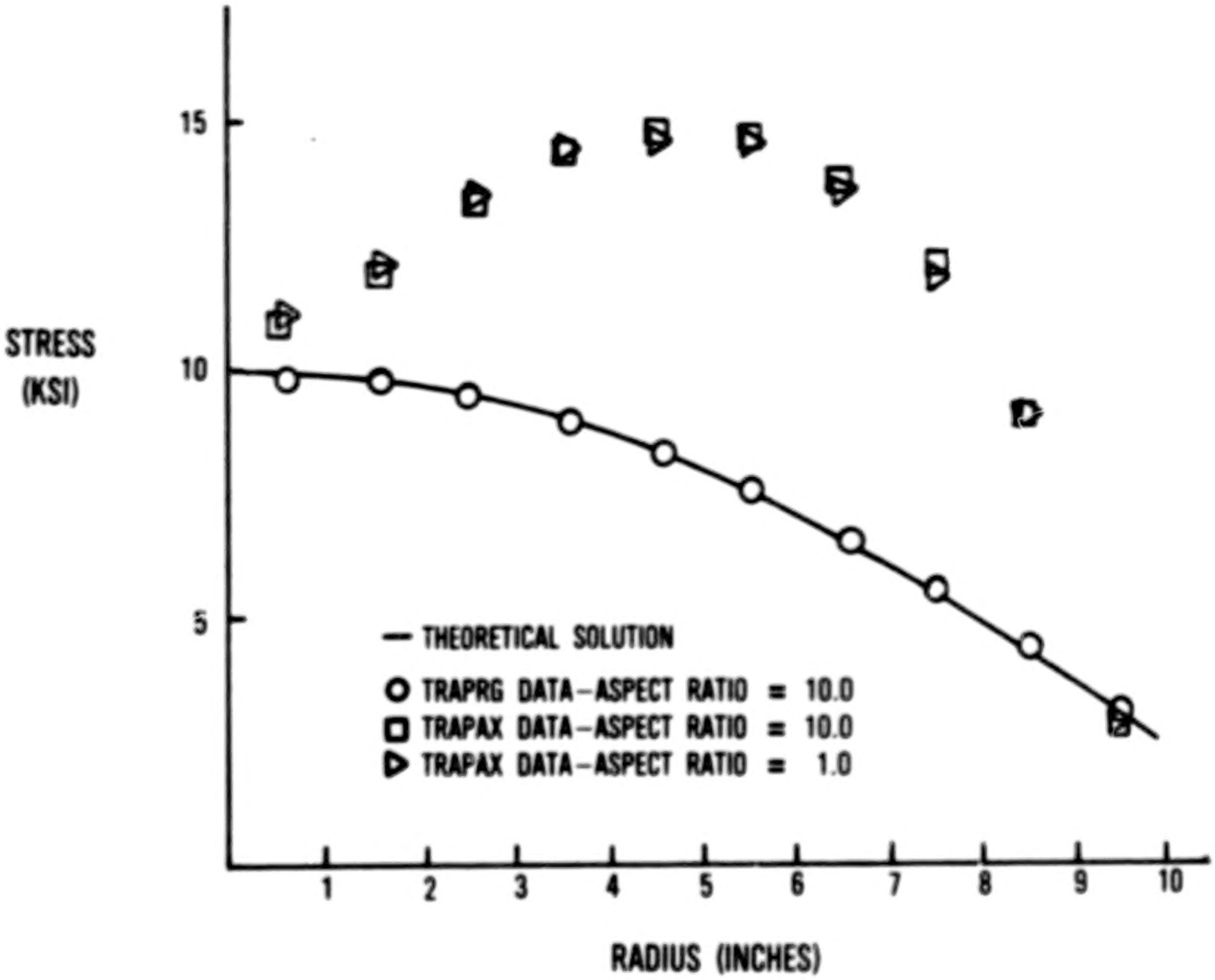
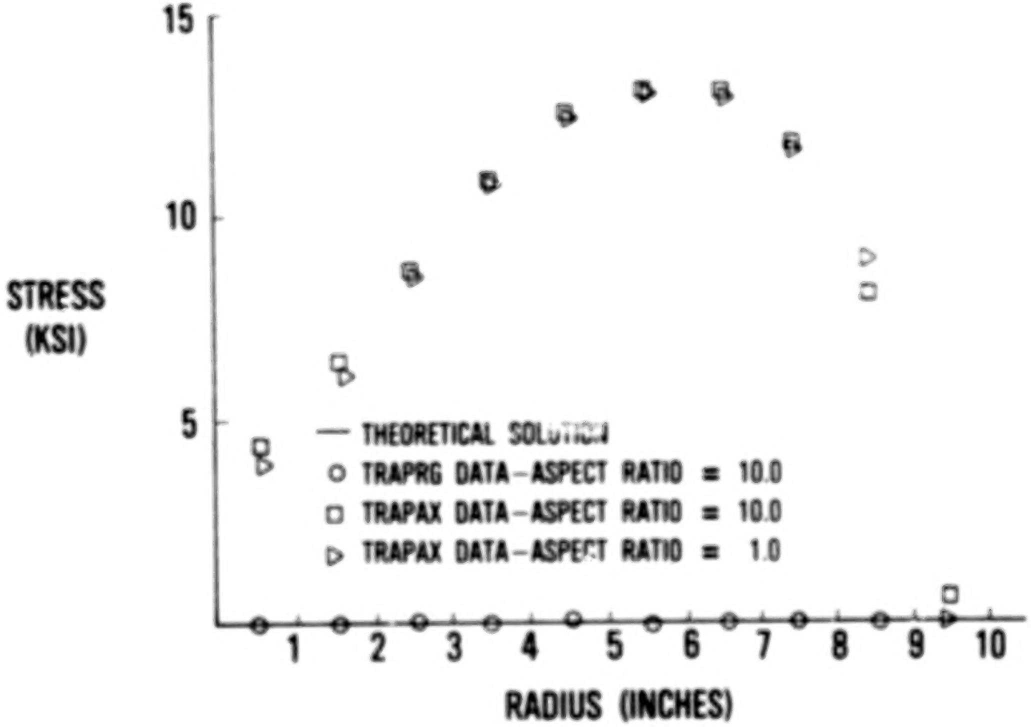

Z
1.0
t/4
r
MODEL WITH RECTANGULAR ELEMENTS
Z
1.1
1.0
.9
t/4
r
MODEL WITH SKEWED ELEMENTS

FIGURE 8. HOOP STRESS FOR SKEWED ELEMENTS (TEMPERATURE INDEPENDENT MATERIALS)



FB 79106-4

FIGURE 9. AXIAL STRESS FOR SKEWED ELEMENTS
(TEMPERATURE INDEPENDENT MATERIAL)



CONTENTS

Page

FOREWORD . . . . [illegible] 1/A5

1. ON THE CONGRUENT FEATURE IN NASTRAN 1 1/A9
by P. R. Pamidi (Computer Sciences Corporation)

2. DEVELOPMENT OF THE [illegible] FOR NASTRAN ANALYSIS. 11 1/B6
by Robert J. Borucha (Gates [illegible] Corporation)

3. TENSIONING OF A BELT AROUND A DRUM USING MEMBRANE ELEMENTS 35 1/E1
by C. H. S. Chen (B. F. Goodrich Company)

4. THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS SUBJECTED TO ASYMMETRIC THERMAL LOADS 41 1/E12
by Gordon R. Negaard (ANAMET Laboratories, Inc.)

5. STRESS CONCENTRATIONS IN SCREW THREADS. 65 2/B1
by G. Peter O'Hara (U. S. Army Armament Research and Development Command)

6. CONDENSING LOADED POINTS FOR TRANSIENT [illegible] SUBSTRUCTURING. 79 2/C6
by Thomas G. Butler (Butler Analyses)

7. IMPLEMENTATIONS OF A TRAPEZOIDAL RING ELEMENT IN NASTRAN FOR ELASTIC-PLASTIC ANALYSIS 107 2/E5
by P. C. T. Chen and G. P. O'Hara (U. S. Army Armament Research and Development Command)

8. NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT 113 2/F5
by Carl A. Meyer (Planning Research Corporation)

9. STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED PERIODIC STRUCTURES 131 2/G9
by Anthony J. Kalinowski (Naval Underwater Systems Center)

10. APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS AT NORTHROP 165 3/C4
by Ashok K. Singh (Northrop Corporation)

11. IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS OPERATING SYSTEM. 189 3/F2
by Stephen S. Britten and Betsy Schumacker (M.I.T.)

MICROFILMED FROM BEST AVAILABLE COPY

Page

12. IFEMS - AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM 205
by Spencer McKellip, Todd Schuman, and Spencer Lauer (Sikorsky Aircraft) 3/G5

13. APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL. 223
by James L. Rogers, Jr. (NASA Langley Research Center) 4/B6

FIGURE 10. HOOP STRESS FOR SKEWED ELEMENTS (TEMPERATURE DEPENDENT MATERIAL)

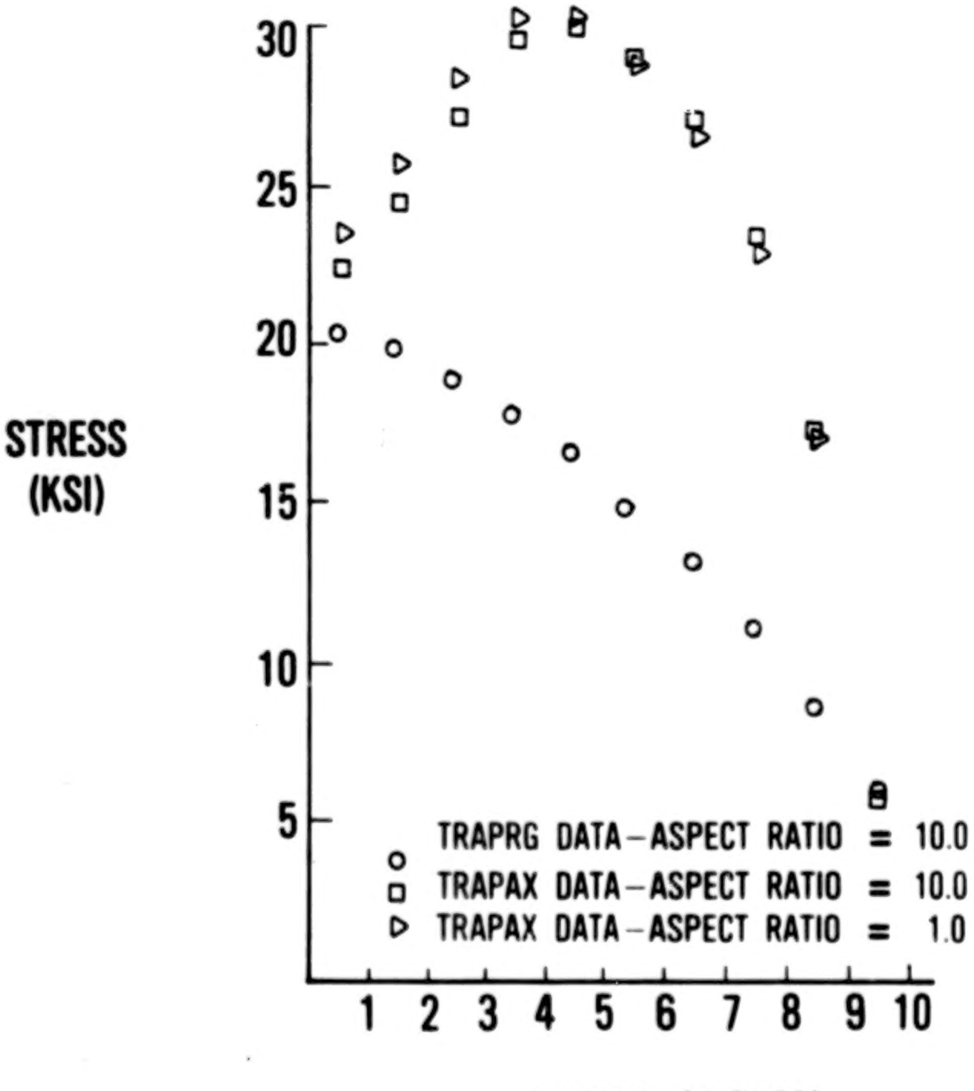
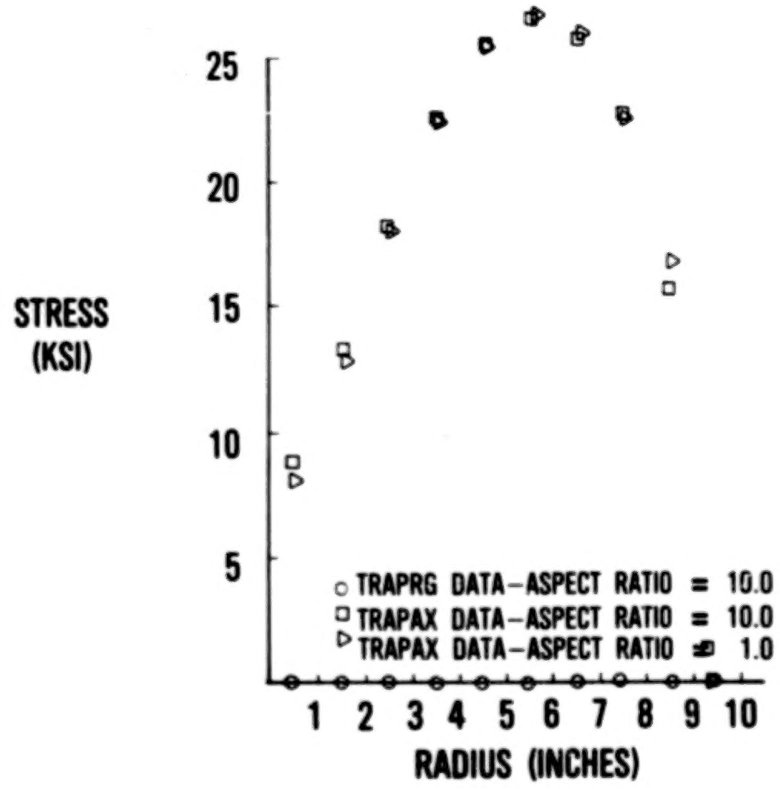
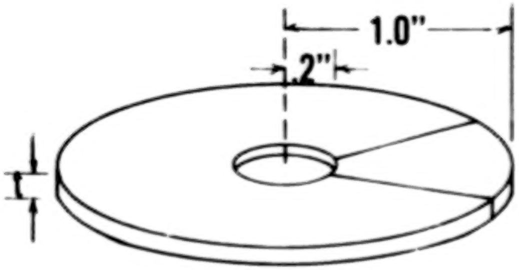


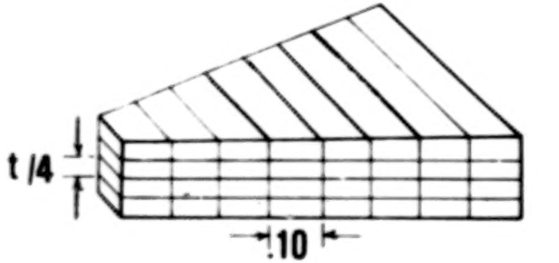
FIGURE 11. AXIAL STRESS FOR SKEWED ELEMENTS (TEMPERATURE DEPENDENT MATERIALS)



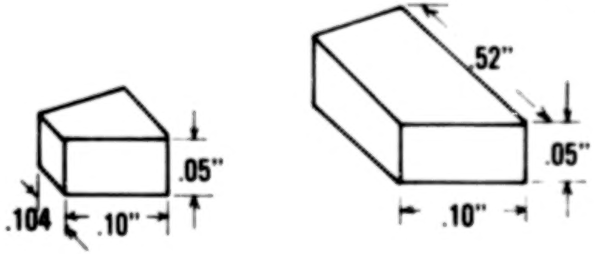
FB 7-9-103 --4

FIGURE 12. AXISYMMETRIC RING MODEL SUBJECTED TO ASYMMETRIC HEATING



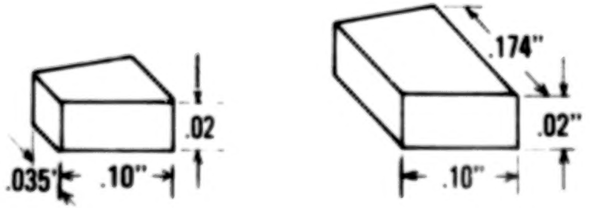


CIHEX WEDGE MODEL USED FOR CYCLIC SYMMETRY


.05"
.104
.10"
.52"
.05"
.10"

30° WEDGE-ASPECT RATIO RANGE

2/1 TO 10/1


.02
.035"
.10"
.174"
.02"
.10"

10° WEDGE-ASPECT RATIO RANGE

5/1 TO 9/1

FIGURE 13. HOOP STRESS IN THE ASYMMETRICALLY HEATED RING (RADIUS = .25 INCHES)

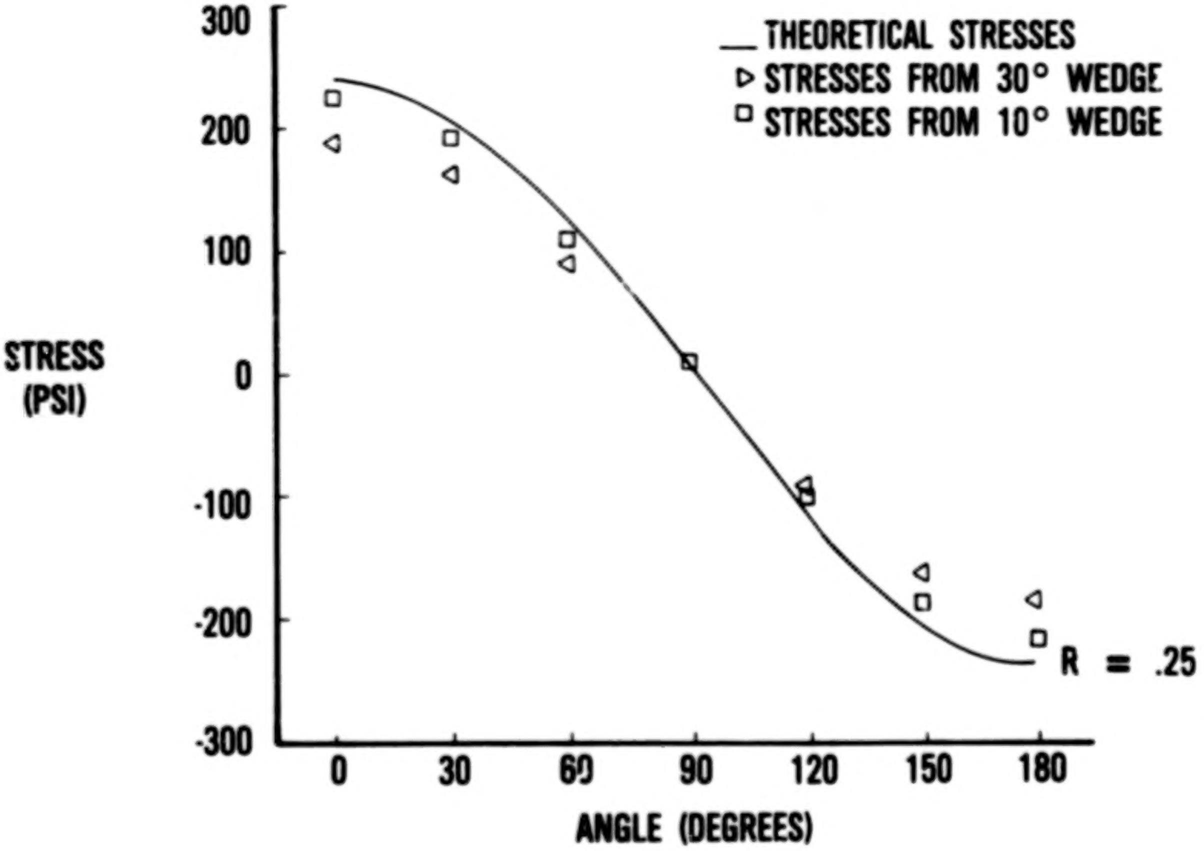


FIGURE 14. HOOP STRESS IN THE ASYMMETRICALLY HEATED RING (RADIUS = .55 INCHES)

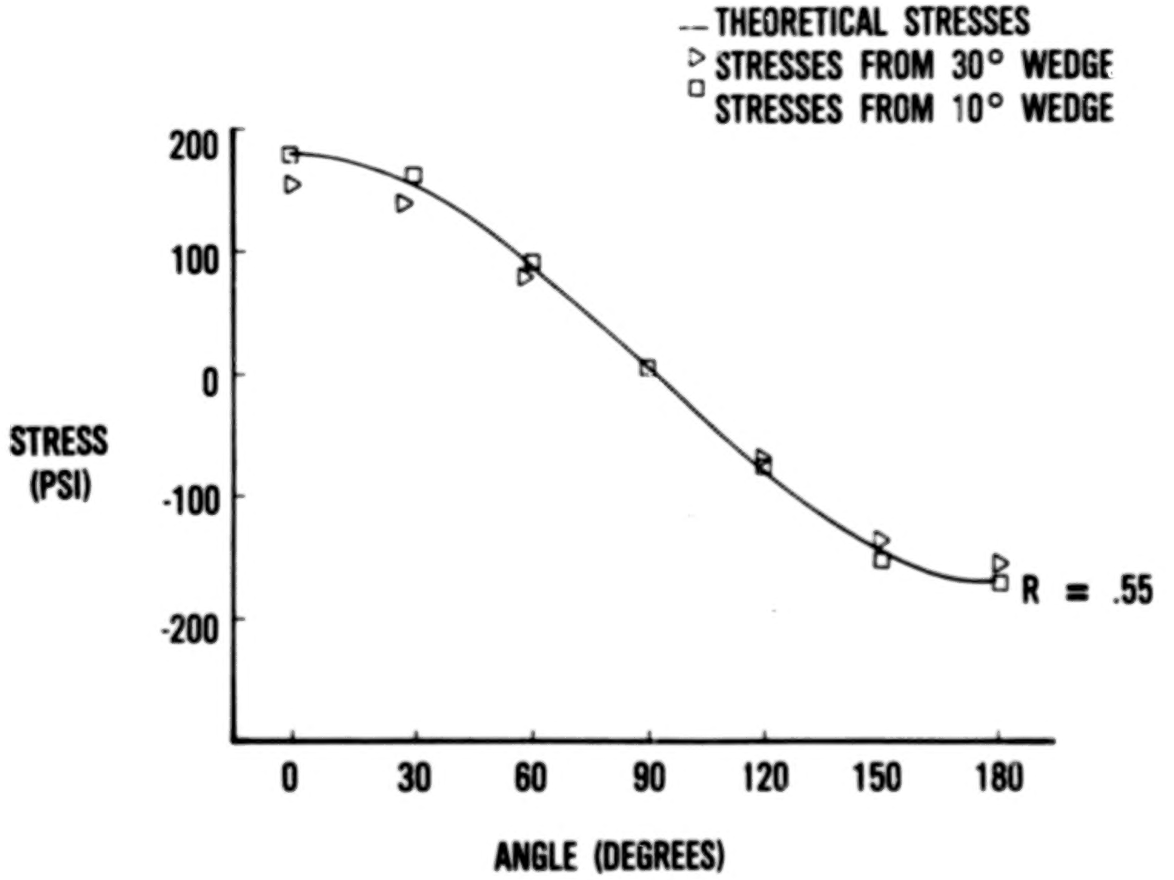
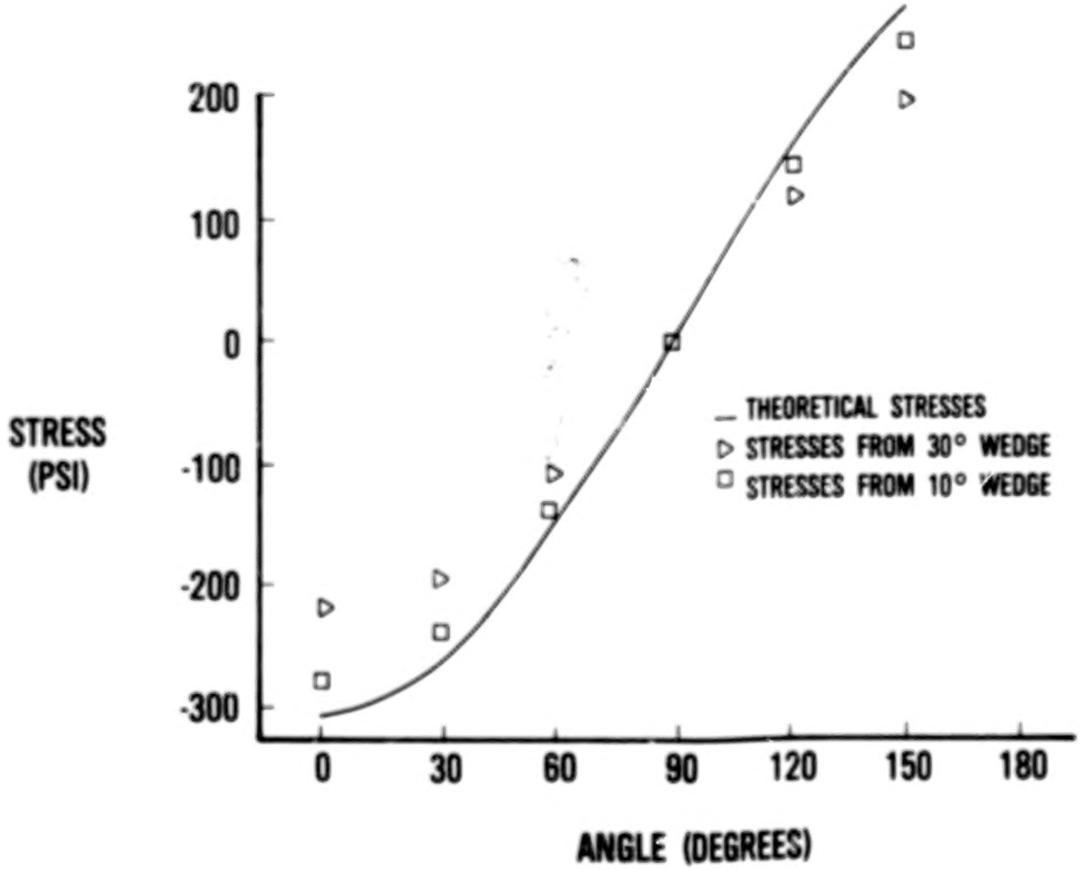


FIGURE 15. HOOP STRESS IN THE ASYMMETRICALLY HEATED RING (RADIUS = .95 INCHES)



BLANK PAGE

STRESS CONCENTRATIONS IN SCREW THREADS

G. Peter O'Hara
US Army Armament Research and Development Command
Benet Weapons Laboratory, LCWSL
Watervliet Arsenal, Watervliet, NY 12189

SUMMARY

The concept of stress concentration in screw threads is defined as a ratio of maximum fillet stress normalized to shear transfer rate. The data is presented as a plot of fillet stress vs. radial stress for a particular thread form. The Heywood equation is used to generate the basic plots and NASTRAN is used to extend the analysis to the case both where flanks of an individual thread tooth are in contact and the case where a finite axial stress is superimposed.

INTRODUCTION

The concept that stress or flow lines concentrate around various structural discontinuities is very old and has been the subject of many thousands of books and technical papers. It is convenient to express this concept in terms of a stress concentration factor (K) using the simple equation:

$$\sigma_{max} = K\ \sigma_{non}$$

Where K is a ratio between the maximum stress and some nominal stress, the single book "Stress Concentration Factors" by R. E. Peterson (ref. 1) is a compilation of the work included in some 378 references. The bulk of this work is contained on graphs which are plots of K vs. some geometry factor and most use a family of curves to show the effect of some other geometry factor. These plots provide both useful numeric information and a quick visual picture of the structural response.

The concept of stress concentration in screw threads is rather elusive and in fact there is little work done on stresses in threads. R. B. Heywood (ref. 2) published an empirical equation for the maximum fillet stress which was used in the work of Weigle, Lasselle and Purtell (ref. 3) as a guide in trying to improve fatigue life of cannon breech rings. Later this author demonstrated that the Heywood equation would give accurate numeric data when the boundary conditions were closely controlled (ref. 4).

However, most work with screw threads seems to be done for specific cases such as the fine work of M. Heyinyi (ref. 5) who investigated bolt shank and nut design in Witworth threaded bolts. This type of analysis using three dimensional photoelasticity was also used by W. F. Franz (ref. 6) and J. D. Chalupnik (ref. 7). A further attempt at optimizing a thread form was done by

R. L. Marino and W. F. Riley (ref. 8).

In all of these works the calculated stress concentration factor is different for each thread in the system. It would seem that if the stress concentration factor is properly defined something should be a constant for all threads of a specific shape. In his original paper, Heywood demonstrated part of the problem. The stress in the fillet is the result of two factors. First, is the stress due to the load on the individual thread tooth and second, is the stress due to the general stress field or the axial stress (σ_a) near the thread fillet. In this paper I will add effect due to friction and normalize all stresses to the average shear transfer rate (τ_R).

When the friction force and the force due to the "wedge" effect of the loaded flank of the thread are combined a radial (normal) force is produced which can be averaged into the radial stress (σ_r). The fillet stress (σ_F) can be expressed as the sum of two functions.

$$\sigma_F/\tau_r = \bar{\sigma}_F = G_1(\alpha,\beta,\bar{R},\theta,\bar{\sigma}_r) + G_2(\alpha,\beta,\bar{R},\theta,\bar{\sigma}_a)$$

In the above equation the first function (G_1) is the relation between fillet stress and the load on the individual thread tooth. The second function (G_2) is the factor due to the general stress field. Alpha (α), beta (β) and $\bar{R}$ are the thread geometry factors. The angle (θ) is in both functions because they do not maximize at the same position in the fillet. In this paper the shear transfer rate is defined as the net load supported by the thread divided by the area at the pitch line. The direction of the net load is parallel to the pitch line and in the analysis this component of the force will be unity. The radial stress ($\bar{\sigma}_r$) and axial stress ($\bar{\sigma}_a$) are normalized to the shear transfer rate.

The above discussion relates to a normal screw thread problem where only one flank of a particular thread contacts one flank of a mating thread. In some structures the relative displacement in the radial direction across the threaded connection is such that the radial gap in the threads is closed and both flanks of each thread carry load. Under these conditions the radial component of the loads add together to produce high negative or compressive radial stress across the joint. The axial loads oppose each other and the pressure on the primary flank must become very high to overcome the secondary flank load. This is not a common condition; however, it may become very important in the cannon breech-to-tube connection.

THREAD GEOMETRY

The thread geometry parameters are shown in Figure 1 and in this report all linear dimensions will be normalized to pitch (P). The primary geometry parameters are the primary flank angle (α), the secondary flank (β) and the root radius $\bar{R}$. These, in conjunction with the pitch space (P1), define the basic thread geometry. Other factors are required to insure a practical thread which will fit together. The addendum (AD) and dedundum (DD) dimensions sum to the total height (HT). The tip radius (RT) eliminates a sharp corner and helps to support the bearing surface. The root flat (FLAT) is often used to make up for the root radius tolerance. The bearing height (Z) is used to calculate the average bearing stress and the shear length (S) is used to calculate the maximum shear-out failure load.

This complicated system is simplified by the fact that we must deal with a small set of standard thread forms. In this report detailed analysis has been done on the British Standard Buttress thread and Heywood analysis has also been done on the controlled root bolt thread or "J" thread and the Watervliet Arsenal Buttress used on cannon breeches. The nominal dimensions for these threads are shown in Table I.

TABLE I. THREAD GEOMETRY DEFINITION

	British Buttress	Watervliet Buttress	30 "V" (Rolled)
α	7°	20°	30°
β	45°	45°	30°
$\bar{R}$	0.1205	0.1333	0.1804
P1	0.500	0.5276	0.500
HT	0.5059	0.4787	0.6077
RT	0.00	0.0480	0.1083
FLAT	0.0	0.0	0.0

LOADING PARAMETERS

The Heywood load parameters are also shown in Figure 1. They are a point force (W) applied at some position (b) in the loaded flank with a friction angle (γ). This scheme can be repeated many times on the loaded flank to produce some load distribution curve. The following loading assumptions are made:

1. The total load vector parallel to the datum line is unity.
2. The load distribution is uniform.
3. Friction does not vary along the flank.

The first assumption given allows the normalization of stresses to shear transfer rate and the other two establish a simple loading case.

Under conditions of high radial compressive load, it is possible for threads to be pushed together until both flanks contact. This condition will be discussed later. Under normal conditions only the primary flanks contact on the thread and the radial stress become a function of the flank angle α and the friction angle γ:

$$\bar{\sigma}_r = \tan (\alpha - \gamma)$$

Note that friction becomes a signed variable depending on the relative displacement of the two components of the structure.

In the above discussion the general field or axial stress is assumed to be zero. In the NASTRAN finite element analysis the axial stress was simulated by the use of a constraint subcase in which the relative axial displacement between the two radial boundaries was fixed by the use of scalar points and multipoint constraint equations. The radial displacements on these planes were made equal for congruent points. The radial displacement of the inner axial boundary was set equal to the Poisson contraction of a solid bar.

HEYWOOD ANALYSIS

The Heywood equation is shown in Figure 2. This is a semi-emperical equation that was fit to a large body of photoelastic data. It calculates maximum fillet stress for a point load on the primary flank of a thread using a specific friction angle. In order to simulate a uniform load distribution, the results are averaged for seven different "b" values evenly distributed over the flank. The process has been programmed into a program called HEY40. The calculations have been done for many standard thread forms, and the three reported in Figure 3, have been defined in Table I. This plot of fillet stresses plotted against radial stress will be referred to as the "thread characteristic curve". This curve covers a friction angle range of -45° to 45° or a coefficient of friction range of -1.0 to 1.0.

In Heywood's photoelastic experiments he was careful to transfer the load supported by the threads profiles in a shear mode to make the axial stress as small as possible. This process limited his equation to the case where axial stress is equal to zero.

NASTRAN FINITE ELEMENT ANALYSIS

The finite element work was done for three reasons: (1) to verify the Heywood analysis; (2) to examine the two-flank problem; and (3) to include a finite axial stress. The grid for the British Standard Buttress is shown in Figure 4. It contains 216 triangular ring elements (CTRIARG) and 133 grid points. The run required five basic loading subcases plus fourteen subcase combinations for each value of axial stress. These fourteen subcases cover both 1-flank and 2-flank contact over a range coefficient of friction of -1.0 to 1.0 in increments of 0.25.

The grid was generated using IGFES (ref. 9) and following that, force sets were calculated to apply uniform pressure and uniform shear loads on both flanks of the thread and a displacement was calculated for a nominal 1.0 psi axial load on the grid. Two different constraint conditions were required to complete the boundary conditions for a single thread taken from a long series of threads. For loads on the thread the inner boundary points were fixed in both radial and axial directions and similar points on the two radial boundaries were constrained to equal displacements. In this way the net load was taken out as shear load on the inner axial boundary and the multipoint constraint equations replaced adjoining material. For the axial load condition the inner axial boundary was constrained to the Poisson displacement in the radial direction and left free in the axial direction. The two radial boundaries were given fixed relative axial displacements and the radial displacement was made equal for similar grid points. This condition was set to simulate a far removed axial loading.

Because the basic loads were all for a 1.0 psi uniform applied pressure or shear and the results were desired for a 1.0 psi shear transfer rate (calculated at the datum line), it became necessary to calculate the correct Subcase Sequence Coefficients for fourteen subcases for each of four axial stress values (or 56 sets). Therefore a small program was generated to supply all necessary SUBCOM, SUBSEQ and LABEL cards for that portion of the case control deck.

Uniform increments of 0.25 in coefficient were used from -1.0 to 1.0. If the friction value was positive, a single subcase combination was generated which would superimpose the two primary flank loads with the proper value of axial stress. If the friction was zero or less, a similar subcase combination was generated along with one where both flanks were loaded and the second radial stress was 1.0 greater than the initial.

The axial load subcase produced the conventional fillet stress concentration factors of K = 2.89. This maximum stress was in an element at the bottom of the fillet where θ is approaching 0°. In the cases where the load is applied to the thread the exact position of the stress maximum is about 45° up from the bottom of the fillet. Data is reported here for two cases of axial stress. Zero axial stress is shown in Figure 5 and axial stress of 2.0 is shown in Figure 6. These plots are a set of six lines with the single contact curve at the right. Starting at that curve is a family of five lines going to the left which represent two flank contacts at different values of friction from 0 to -1.0.

DISCUSSION

The first thing to note is that there is excellent agreement between Heywood and NASTRAN over most of the range of the plots for the one case in question. Because of this, the Heywood relation can be used to evaluate different thread forms. The finite element method has allowed the expansion of the basic plot to the 2-flank contact problem and the addition of the axial stress.

There are several important points that are demonstrated by this work. Note that a small change in friction can produce a large fillet stress variation in all three threads reported in the Heywood analysis. Negative friction angles can produce marked reductions in thread fillet stress. This effect was noted several years ago in an unpublished three dimensional photoelastic study where the model was overloaded and the threads were forced to a high negative radial stress. In this case the fillet stresses were very low and the experiment was repeated. This author suspects that friction variation may be responsible for much of the scatter in bolt-fatigue data.

This work was initiated becuase of the necessity of analyzing a structure with a long threaded connection using many small threads. In this case the modeling of each thread would require an excessively large data deck. Therefore, the threads were handled as a conventional contact problem where friction could take on any value and limits were applied to the radial stress. In the solution the contact surface was placed on the datum line of the threads and one or two teeth were replaced by one element space. Shear transfer rates could then be estimated from the shear stress data near the contact surface along with radial and axial stress. The fillet stresses were estimated for use in fracture mechanics analysis.

CONCLUSION

A stress concentration approach to the thread fillet stress problem has been defined using the shear transfer rate as the fundamental quantity. This stress concentration is plotted for a fixed geometry in a stress vs. stress plot where the stress concentration is a function of the applied radial stress. This process can be repeated for several values of the applied axial load. The effects of axial stress and applied thread loads seem to be of equal importance and accurate results require the analysis of both factors.

REFERENCES

1. "Stress Concentration Factors," R. E. Peterson, John Wiley & Sons, New York, 1974.

2. Heywood, R. B., "Tensile Fillet Stresses in Loaded Projections," Proceedings of the Institute of Mechanical Engineers, Vol. 160, p. 124, 1960.

3. Weigle, Lasselle and Purtell, "Experimental Behavior of Thread-Type Projections," Experimental Mechanics, May 1963.

4. O'Hara, G. P., "Finite Element Analysis of Threaded Connections," Proceedings of the Fourth Army Symposium on Solid Mechanics, September 1974.

5. Hetinyi, M., "The Distribution of Stress in Threaded Connections," Proceedings of SESA, Vol. 1, No. 1, 1943.

6. Franz, W. F., "Three-Dimensional Photoelastic Stress Analysis of a Threaded Pipe Joint," Proceedings of SESA, Vol. , No. , 1950.

7. Chalupnik, J. D., "Stress Concentration in Bolt Thread Roots," Experimental Mechanics, 1967.

8. Marino, R. L., and Riley, W. F., "Optimizing Thread-Root Contours Using Photoelastic Methods," Experimental Mechanics, January 1964, p. 1.

9. Lorensen, W. E., "Interactive Graphic Support for NASTRAN," Sixth NASTRAN User's Colloquium, NASA Conference Publication 2018, October 1977.

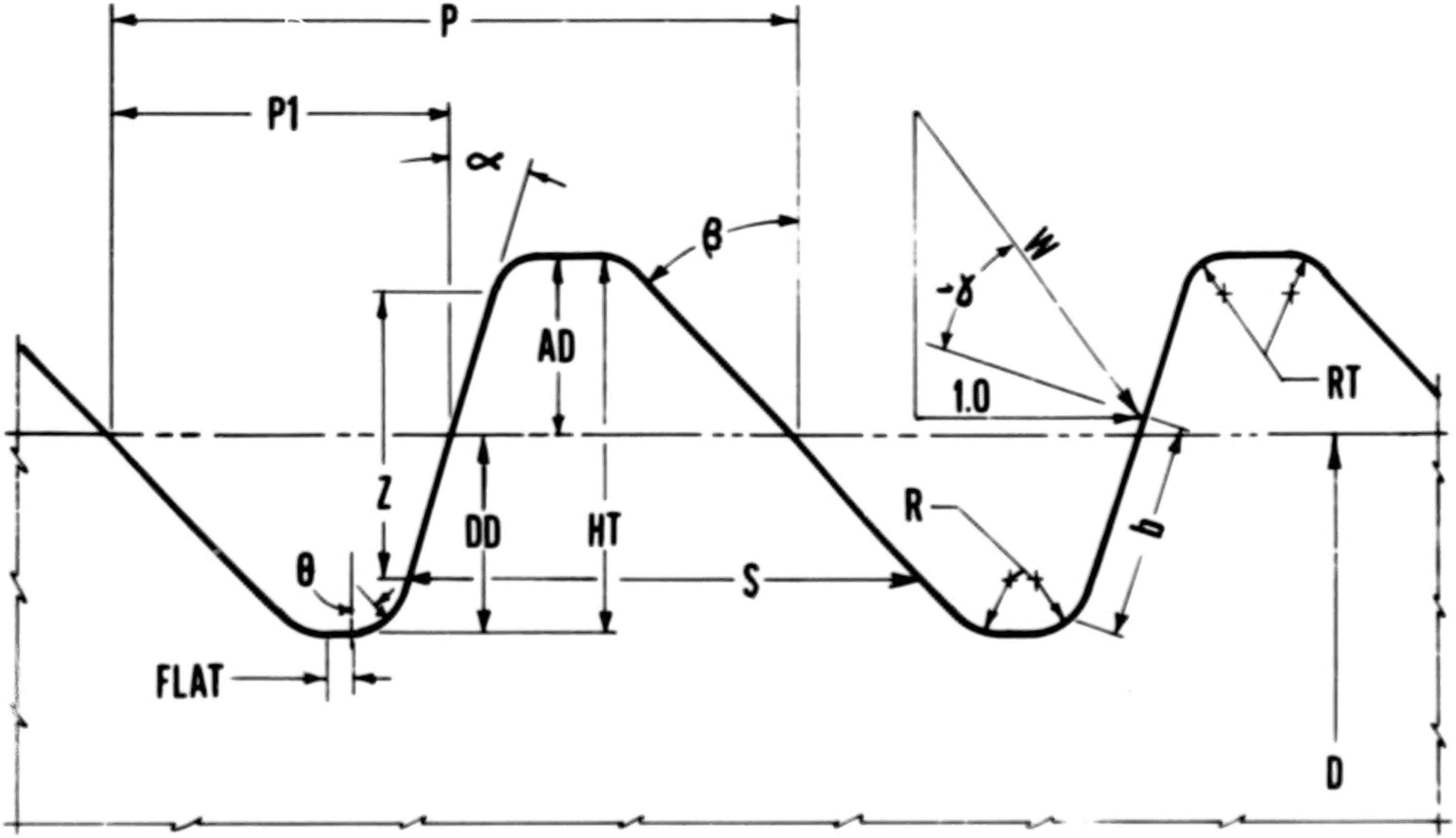

P
P1
α
β
W
-γ
1.0
AD
RT
Z
DD
HT
R
b
θ
S
FLAT
D

Figure 1. THREAD GEOMETRY AND LOAD PARAMETERS

$$\sigma_F = \left[1.0 + 0.26\left(\frac{e}{R}\right)^{0.7}\right]\left[\frac{1.5}{e^2} + \sqrt{\frac{0.36}{be}}\ (1.0 + \tfrac{1}{4}\,\text{SIN}\,\gamma)\right]\frac{W}{T}$$

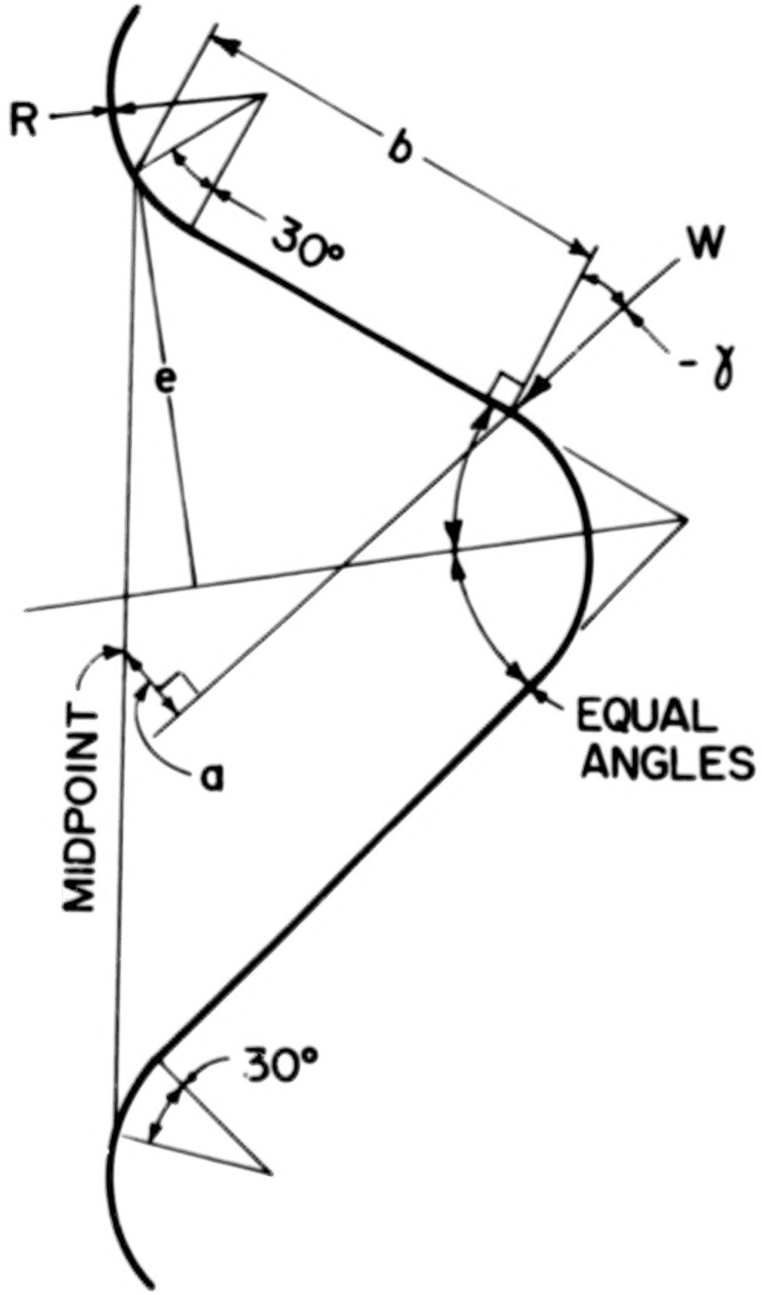

R
b
30°
W
- γ
e
EQUAL ANGLES
a
MIDPOINT
30°

Figure 2. HEYWOOD'S EQUATION

#2

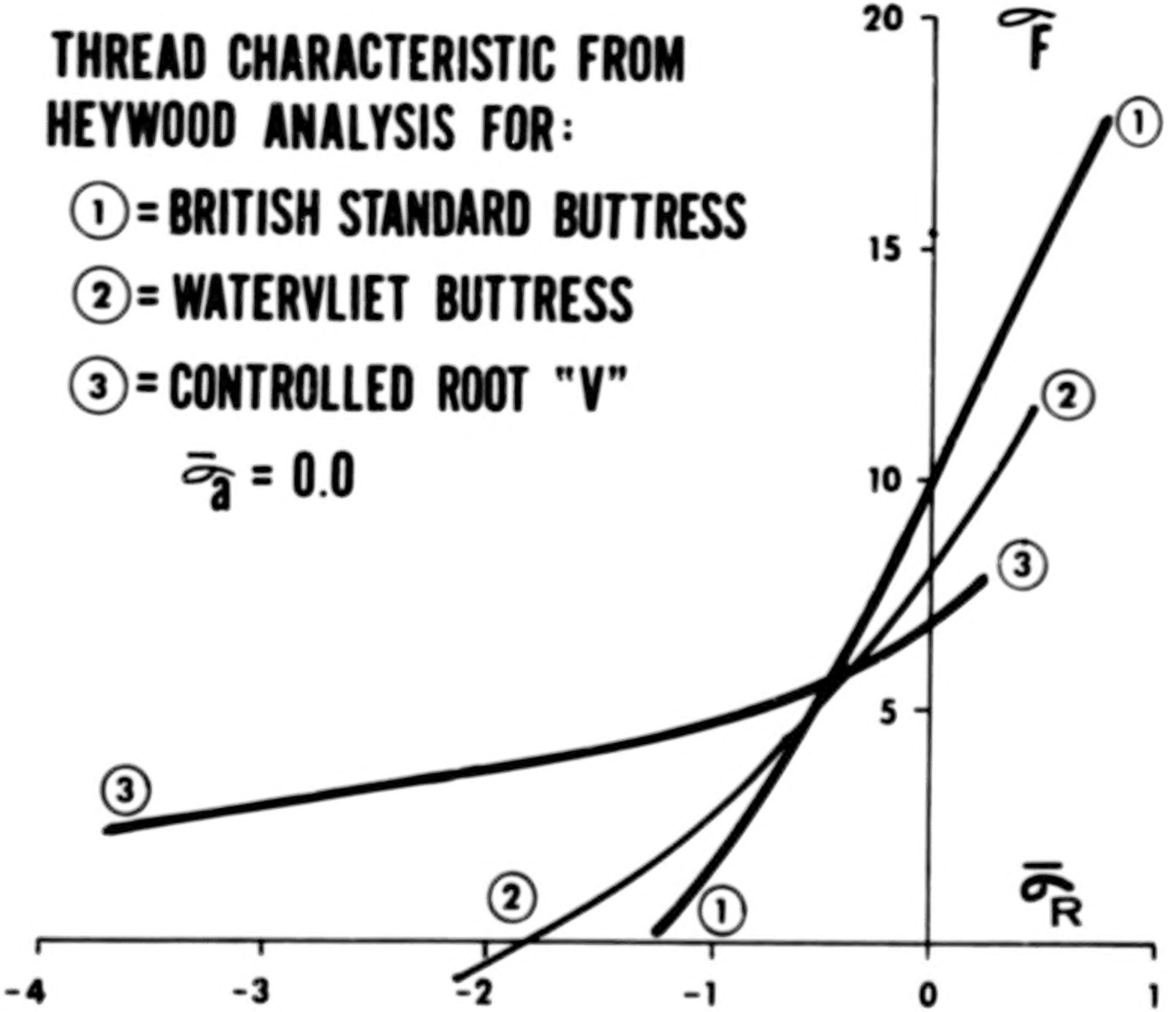

THREAD CHARACTERISTIC FROM HEYWOOD ANALYSIS FOR:
① = BRITISH STANDARD BUTTRESS
② = WATERVLIET BUTTRESS
③ = CONTROLLED ROOT "V"
$\bar{\sigma}_a = 0.0$
$\bar{\sigma}_F$
$\bar{\sigma}_R$
20
15
10
5
-4
-3
-2
-1
0
1
①
②
③

Figure 3.

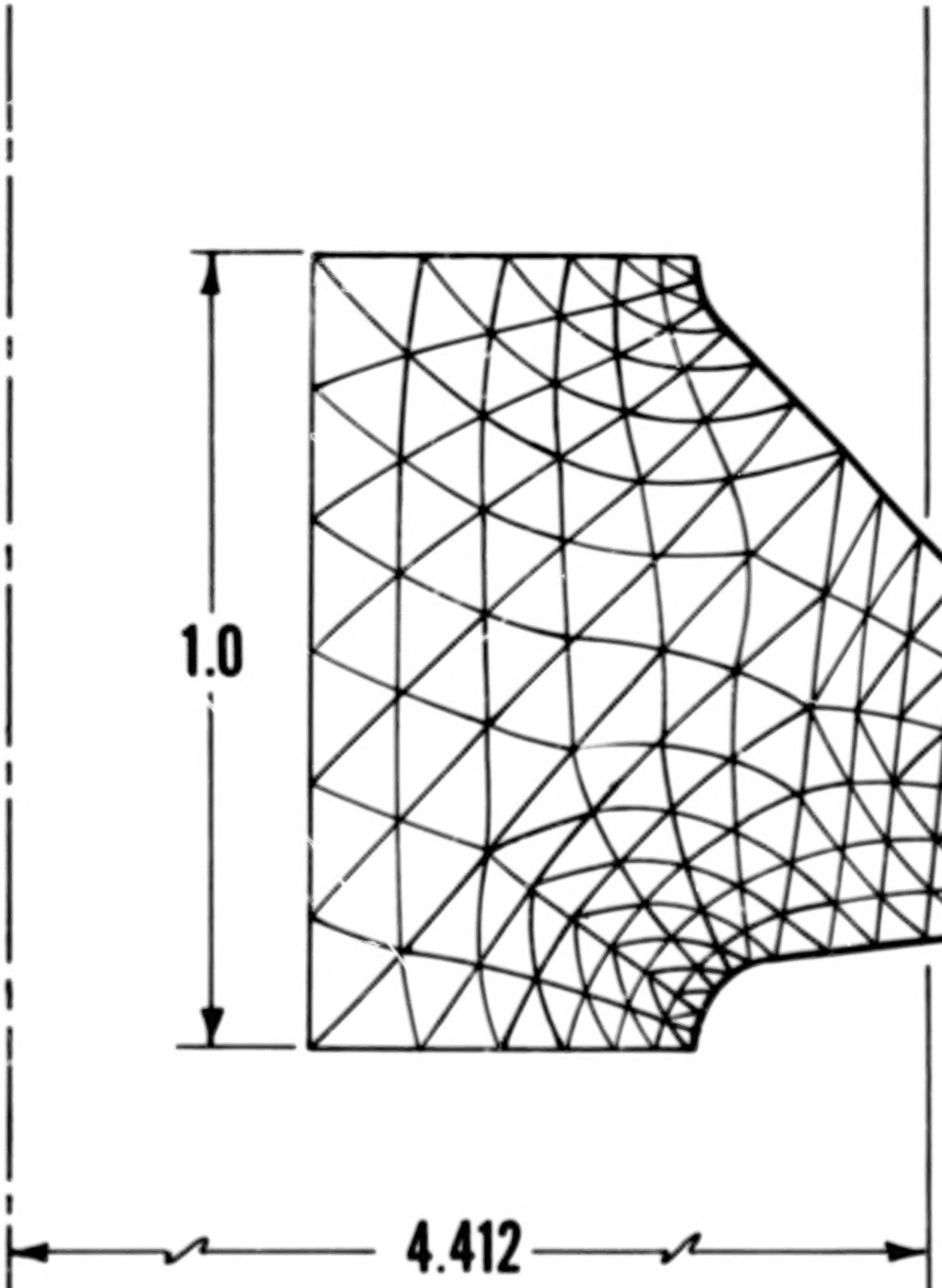
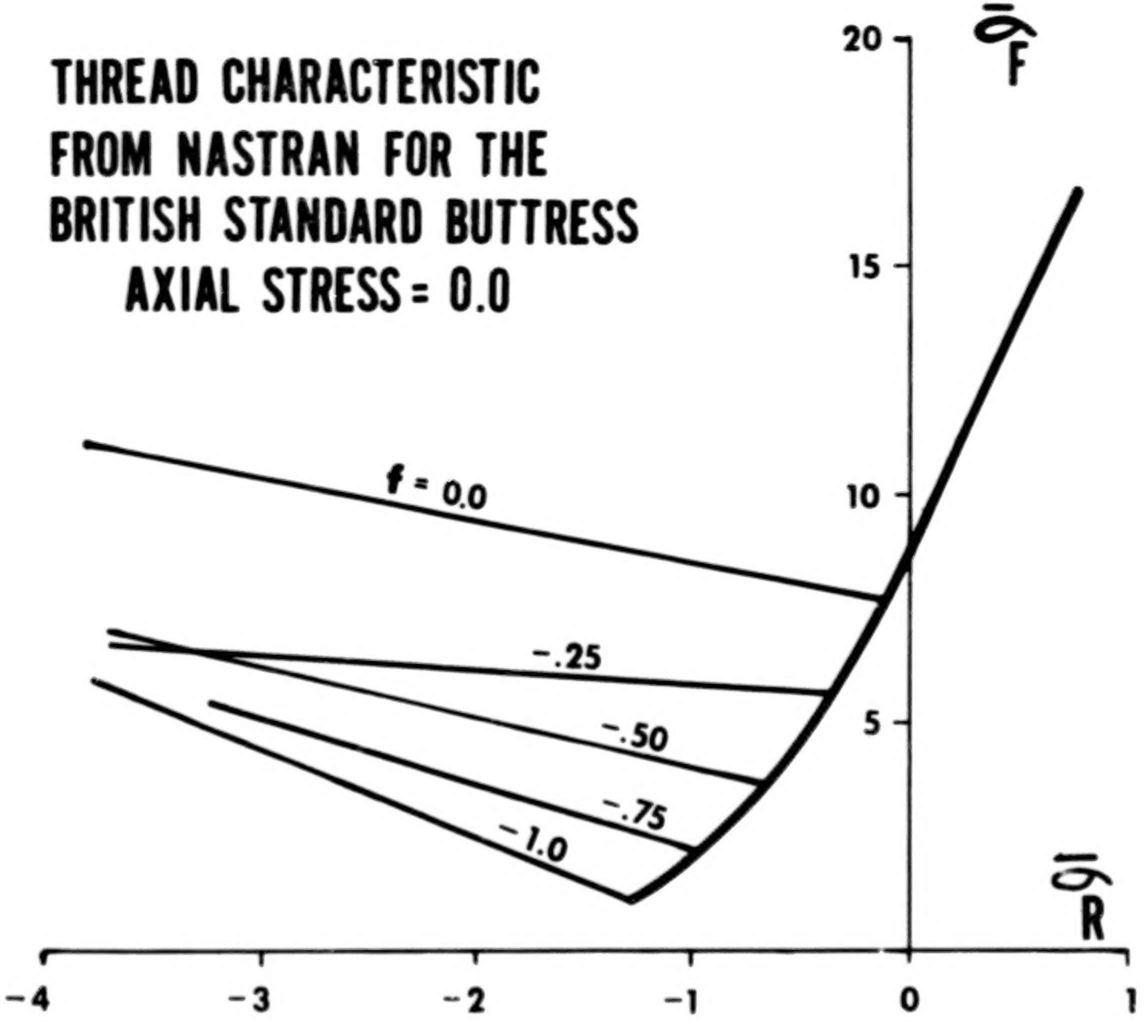


Figure 4. NASTRAN GRID

BLANK PAGE

BLANK PAGE



THREAD CHARACTERISTIC
FROM NASTRAN FOR THE
BRITISH STANDARD BUTTRESS
AXIAL STRESS = 0.0

Figure 5.

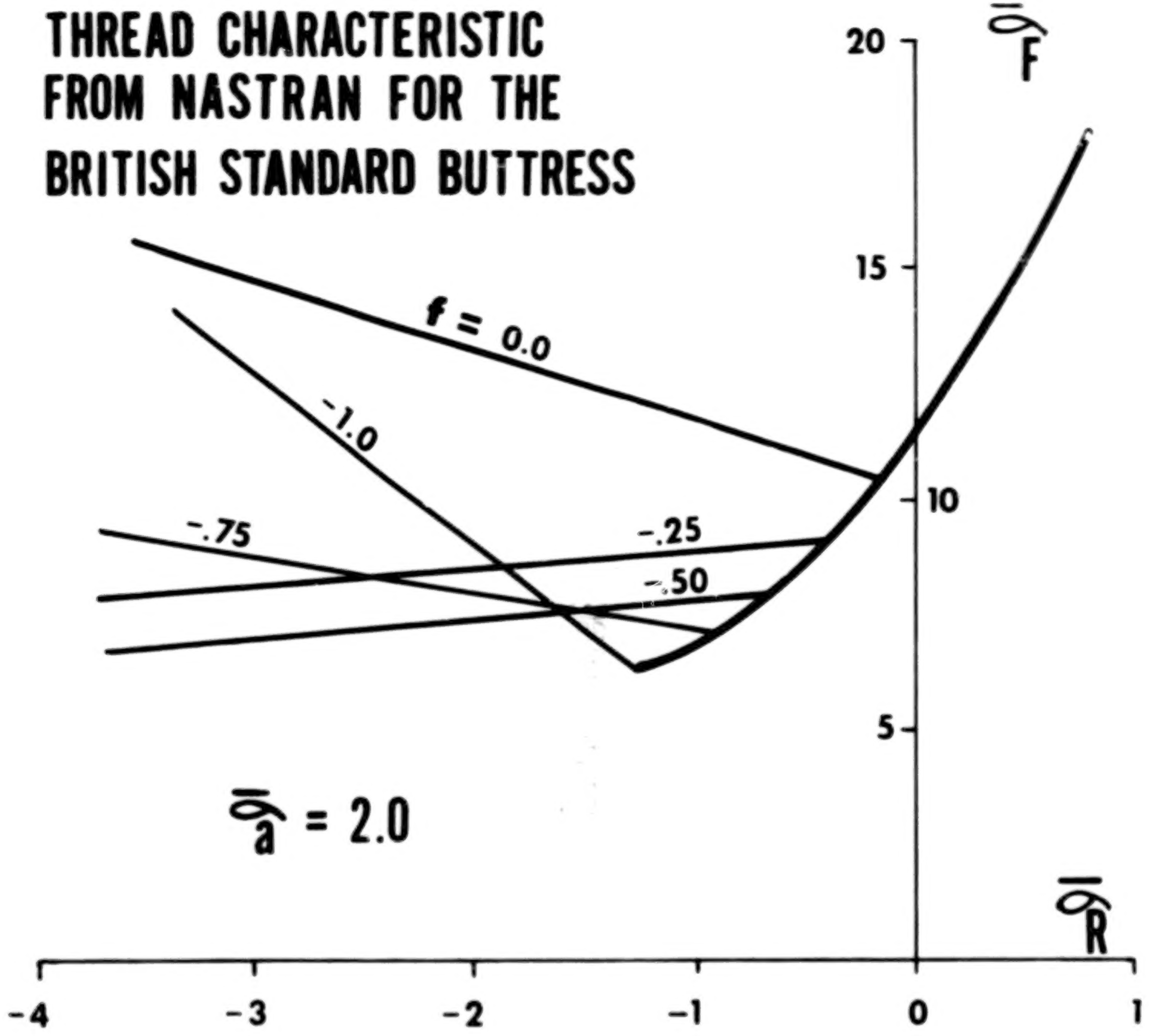

THREAD CHARACTERISTIC
FROM NASTRAN FOR THE
BRITISH STANDARD BUTTRESS
$\bar{\sigma}_F$
20
15
10
5
f = 0.0
−1.0
−.75
−.25
−.50
$\bar{\sigma}_a = 2.0$
$\bar{\sigma}_R$
−4
−3
−2
−1
0
1

Figure 6.

BLANK PAGE

CONDENSING LOADED POINTS FOR 'TRANSIENTS BY SUBSTRUCTURING'

THOMAS G. BUTLER

BUTLER ANALYSES

SUMMARY

This paper is an extension to my article presented at the 7th NASTRAN Colloquium entitled "TRANSIENTS BY SUBSTRUCTURING". In order to capitalize on the economies of (a) condensing of omitted points during substructuring, and (b) of applying BANDIT at the only place where it can enter the analysis, a strategy needed to be worked out so that dynamically loaded points could also be condensed in Phase 1. This would obviate having to retain points during dynamic analysis for no other reason than to make them available for assigning load, and having to suffer the consequences of expensively large matrices. This paper describes a technique, vintage 17.5, for accomplishing these aims. A number of problems arose such as transferring of condensed loading information from R. F. 1 to R. F. 9, and giving this loading a correct time history. The method has been applied to substructure transient solutions according to the approach as outlined at the 7th NASTRAN Colloquium. Only DMAP statements and some manual transfer of data were used to implement this strategy.

PURPOSE

The cost of solving large order transient problems can be high. If the matrix order is reduced by condensation without inflating the band, the cost of solving transients can be decreased. The cost of condensing in itself can be expensive for large matrices, so it behooves one to condense at the component level whenver the matrices are small and the cost is exponentially less. If this condensing of dynamically loaded degrees of freedom could be worked out, then the rder and band of the matrices in the transient solution could also come under the control of the analyst to conform to his needs without operational constraints. The avenue is open in substructuring to do the condensing of each low order component matrix for small cost with a net overall saving. It appears that the ideal is at hand to cut the cost of transients without incurring high processing costs, but the way is not straightforward. The plan should not be coded without considering the needs of banding. One is compelled to condense at the compoonent level when possible, because BANDIT can be invoked in substructuring only at the component level. The purpose of this enterprise is to find a way of condensing dynamic loads in stubstructuring economically, so that the load matrices can be correctly formed when R. F. 9 is entered.

PROBLEMS

How can the dynamic loads be represented in Phase 1? How can the spatial, time amplification, and time delays be coordinated, if the points scheduled for condensation have loading times that differ from those of retained points? How can loading on the retained points be merged with that resulting from condensing operations on the omitted points? Can all these objectives be met without burdensome labor or cost? The answers to these questions will be taken up in the section entitled IMPLEMENTATION. The reason for having to pose the first question is that Phase 1 of fully automated substructuring is operational for statics and eigenvalues only and not dynamics. Consequently, some property of the dynamic load has to be isolated which behaves in a way that can be treated in statics. The answer to the second question seems to be even more elusive than the first. It appears like an attempt to attach a time variation to something which has disappeared from the problem. The merging difficulty posed in the third question seems to be one of retaining some sort of identity of the condensed loads so that the relationships to the loads on uncondensed points can be maintained. The bulk of this report is spent on how these questins were answered.

IMPLEMENTATION

Consider the composition of the dynamic load in its simplest form

$$P(t,x,y,z) = A(x,y,z) \times T(t-p)$$

The component, A(x,y,z), is the spatial specification of the loading. The data for this part is entered via DAREA cards. In that the information contained on DAREA involves position and magnitude, it is similar to stati loads. If one were to set up the TLOAD1 cards for the dynamic loads without regard for any intention to condense loaded points, it would be easy to identify those spatial sets (i.e. DAREA sets) that are loaded synchronously; i.e. those having a common time amplification defined by a single TABLED1 set and a single DELAY set. At completion of defining the total dynamic load there will be as many synchronous sets as there are TLOAD1 sets. There are, at most, the same number of DAREA sets as there are TLOAD1 sets--in general there are fewer DAREA sets. It is now a matter of comparing the omitted degrees of freedom with the DAREA sets to find their intersection. All those DAREA sets having omitted degrees of freedom are isolated. For purposes of illustration, concentrate on just one substructure and assume there are three such DAREA sets in that one substructure and designate them as Q,R & S. Use lower case letters q,r & s for the sets of omitted degrees of freedom in their respective parents. Delete q,r & s from Q,R & S and label these DAREA sets with their condensed d.o.f.'s removed: Q',R' & S'. The case in which all degrees of freedom in one of the DAREA sets are omitted is admissable; for instance q could be identical to Q and Q' could be null. With this notation problem behind us it will ease our discussion of the strategy. These individually synchronous sets (q,r & s) that are both loaded and omitted, will be operated on during substructuring. In effect q,r & s are going to be condensed as static loads in Phase I then delivered back into the dynamic loads as pre-condensed DAREA sets. The logic follows from the fact that spatial condensation in statics operates exactly the same way that spatial condensation works in dynamics. Time varying amplification won't operate any differently on a condensed set that comes pre-condensed from statics and is not further condensed in dynamics than it would on one that comes earmarked to become condensed during transient execution. Consequently, when the loads q,r & s are condensed in substructuring, the resulting redistribution of loads to retained points will be labeled q',r' & s'. The next step is to organize data from q',r' & s' in DAREA format and give them the same set I.D.'s respectively as the parent Q',R' & S' sets. Belonging to the same DAREA set, constitutes a merge of sets in Case Control management, so the q,r, & s that were deleted from Q,R & S will have been replaced with their pre-condensed counterparts q',r' & s'. No changes are made to the dynamic loading data involving TLOAD1, DLOAD, and TABLED1 entries. New DELAY cards will be organized. The sets of synchronously omitted degrees of freedom q,r & s will have to be replaced by the corresponding sets of retained degrees of freedom q',r' & s' without any change to the delay times. When more than one substructure has condensed loading, the same procedure as outlined above should be followed for each component.

So much for talk. It remains to be seen whether this idea has substance or is just a brave show. The DAREA entries of loading coefficients that would have been assigned to the omitted d.o.f.'s of a substructure in a solo transient run are assigned instead to those same omitted d.o.f.'s, as static force loads in a Phase I run of that substructure. Each synchronous set is organized as a static load set and each is scheduled in a succession of subcases. In terms of the symbols of the previous paragraph, the DAREA entries on the set q wil be arranged in a static load set for the first subcase, those on r for the second subcase, and those on s for the third subcase. Of the three solution routes available in automated substructuring, the choice converges on R. F. 2 (Inertia Relief), because it offers options for stiffness, mass, and load matrix generation, while R. F. 1 omits the mass and R. F. 3 omits the static load. If one scans the DMAP sequences for R. F. 2 and compares it with the automated ALTER statements for SUBSTRUCTURE one sees that the load vector is condensed down to retained degrees of freedom in the module SSG2. It is output as the data block PL. NASTRAN needs to be told that all 3 matrices are wanted. In the SUBSCC this is indicated by selecting OPTIONS = K,M,P. Nothing more is essential during Phase I except to emphasize that BANDIT should be enlisted prior to Phase I to resequence all points except mating points at the interfaces between substructures. The user should manually assign the sequence numbers to interface mating points. Even though all the Phase I requirements are met, we are not going to leave it with such an uncaring attitude. Something extra is going to be worked up to give a better "feel" as to how things went during Phase I. We are going to "gin up" an ALTER packet to see what the condensed DAREA components look like before being content with the substructure component runs. The ALTER packet will use the module SOFI to pick off PL, as it is delivered to the SOF, and then will inflate PL with zeroes in a succession of merges until it is G-sized. Then this matrix of "load vectors" is delivered to SDR2 for processing the OLOAD requests in Case Control and reformating according to external grid sequencing. Now the OFP handles the printing chores. Particulars of the DMAP ALTER packet are given in Table I. The above outlined procedure is carried out for every substructure which contains omitted loaded points.

If further condensing is decided on in Phase II, using the REDUCE command, the bookkeeping of the several substructures is kept straight inside the SOF data base. When the Phase II operations have prepared the final pseudo structure, the SOF will contain all of the information that is needed to proceed into transients, provided that OPTIONS = K,M,P has been used for every Phase II run. The stiffness and mass matrices are in final form and the PVEC item of the SOF contains a vector for each synchronous subcase of coefficients for all component substructures in fully condensed form. Now an ALTER uses the module SOFI to take the stiffness, mass, and load data from the SOF and deliver them externally. Use OUTPUTT1 for K & M for subsequent insertion into R. F. 9. Use OUTPUT2 for P if a FORTRAN preprocessor is to be used. If DAREA input is to be prepared manually then the PVERC matrix need only be printed. MATPRN will be sufficient since the DAREA data will be written by internal scalar index number instead of external grid point numbers. Details of the Phase II ALTER are presented in Table II. The highlights of the strategies for the Phase I and Phase II solutions are depicted in Tables V & VI.

EXAMPLES AND CHECK CALCULATIONS

A small two component substructure problem was used to illustrate the method of substructure condensation of dynamic load prior to transient execution. Component BOX is a parallelepiped of BAR elements with BAR diagonals on the rear end to make a connection in the middle. Component END is a two bar appendage. The combined pseudo structure is called ALL. Figure 1 shows P/S ALL with grid points numbered and loading indicated. Grid points 2 and 4 are loaded simultaneously with load F(t) and GP's 6 & 8 are loaded simultaneously with load L(t). GP's 1 & 3 are loaded simultaneously with load F(t) delayed an amount T. GP's 5 & 7 are loaded simultaneously with load L(t) delayed by an amount T. Figure 2 has plots of loadings L(t) and F(t). The direction for both dynamic loads F(t) and L(t) is in component direction 1 as shown by the double shafted arrow in firgure 1. If DAREA and DLOAD coefficients are held at unity then the values from figure 2 will become the TABLED1 entries. This pilot problem was deliberately kept small to test only the new features and not redo all features of transients by substructuring. D.o.f. no. 1 was omitted from point 6 of load L(t) and from point 3 of load F(t) and is indicated with tiny circles in figure 1.

In the spirit of keeping things simple, only displacement responses in the uncondensed component END will be examined as indicated by open brackets in figure 1. Since displacements can be recovered as part of the R.F. 9 output, there will be no need for scheduling Phases 3 & 4. In order to test the method, a separate transient analysis entirely within R. F. 9 was run. The problem flow is diagrammed starting with a Substructure Tree in figure 3 and flow charts of transients first with a substructure preface in figure 4 and then in a stand-alone mode in figure 5.

A measure of the chore of converting the PVEC from substructuring into DAREA data for transients can be obtained by looking at the effects of condensing. Figure 6 shows the DAREA values for the synchronous set of grid points 6 and 8 as it was prepared for the solo transient solution. Figure 7 is a sketch of the points to which the omitted loads of GP 6 and GP 3 were vectored by the SMP module in Phase I. Figure 8 shows the results of manually prepariing the DAREA data of the condensed loading for the synchronous set of grid points 6 and 8 which are now organized according to interior scalar points. It is easy to imagine how burdensome this manual preparation can become for a large structure. A processor program is an obvious necessity. Only one digit of data was used to illustrate the relative magnitudes of the redistribution. The major contributions are circumscribed. Translations are enclosed in boxes and rotations are enclosed in circles. The translation load of GP 6 is essentially reduced to a translation at GP 5 (Scalar dof 7) and rotations about GP 8 (Scalar dof 18) and GP 2 (Scalar dof 35). The box about scalar dof 13 identifies the uncondensed loading of GP 8.

RESULTS

In order to test this method of condensing dynamic loads bv substructuring, a separate analysis was performed in the conventional way by submitting the entire structure with loads defined externally and all other features retained, using R.F. 9 in a solo run. The quality of the substructure analysis was based on making comparisons of only displacements at the tip of the appendage where the amplifications would tend to be the greatest. Comparisons of stresses and forces in elements were omitted for reasons of brevity. Displacement comparisons could be made directly from the two transient outputs, but stresses and forces would have required two extra steps plus additional work in Phase II. Results of the two displacement time histories are assembled side by side for each translational and rotational component in Tables III & IV. Where the displacements are large, the outputs match in 6 digits and start to vary in the seventh place. That is, the axial and vertical translations and the rotations about the transverse axis compare almost exactly. But the small displacements don't even match up in the first digit. In theory all of these extremely small displacements should have been zero, because the load was symmetrical; however the condensation was unsymmetrical. This was a very small model that would be highly sensitive to irregularities, whereas in a large structure much smoothing would take place. This unsymmetrical condensation caused numerical noise to creep into both the solo and the substructure solutions, giving non-zero values 8 orders of magnitude less than the signifigant displacements. Noise is highly dependent on the sequence of operations and the sequences in the two cases under scrutiny were different, so the noise from the two can be expected to be different. The net opinion is that the results compare favorably.

On the basis of good correspondence in the responses from the two approaches, we can say that the technique of spatial condensation of DAREA coefficients by simulating them as static forces is satisfactory. The method of condensing dynamic loads by substructuring has been established.

RECOMMENDATIONS

When transients are automated for substructuring, it appears that the elements contained in this non-automated approach could be used as a basis for the algorithm. The Phase I bulk data could include a DAREA card, called say DAREAS, which would respond to the user's input of condensed loading sets. A caption in the LODS item could append a D to every load subcase that originated from DAREAS input. At transient execution time the user would indicate which sets of DAREAS data would carry the same set I.D. as uncondensed transient DAREA data, and which sets of DAREAS data would be distinct. Similarly, the Phase I bulk data could include a DELAY card, called say DELAYS, which would respond to the user's input of condensed loading. Then the usual preparation of the elements of dynamic loads would proceed as is now done in transients. If further condensation is desired in transients when preceded by substructuring, then caution would have to be exercised in data recovery. Instead of passing the UDVT matrix to Phase IV, the data recovery would have to reconstruct the response histories to the full order of the original Scalar Point compliment, before passing the displacements to Phase IV. These displacements are contained in data block UPV. Modifications to the DMAP ALTER's for R.F.9 as outlined at the 7th Colloquium would be needed to allow for an option to output either UDVT or UPV based on the condition of whether OMIT's are present. The subsequent partitioning would have optional input of either UDVT or UPV.

In effect, the user could be in full command of the condensation of his problem in substructuring by using this proposed technique. He could condense in Phase I where it is most economical and where he could take maximum advantage of BANDIT. He could condense again in Phase II where interface points could be removed. Lastly, he could condense in transients where the final trade-offs are reviewed between the transient solution costs and additional condensation and data recovery costs.

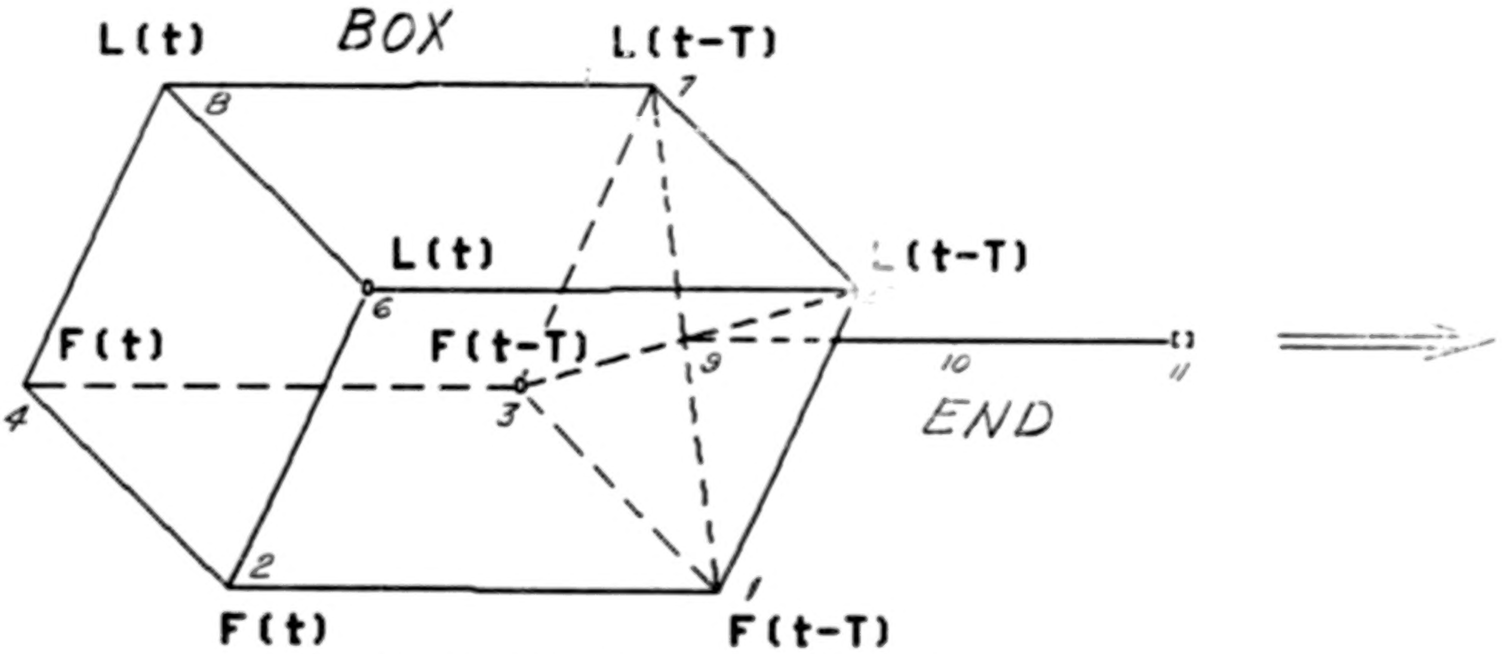
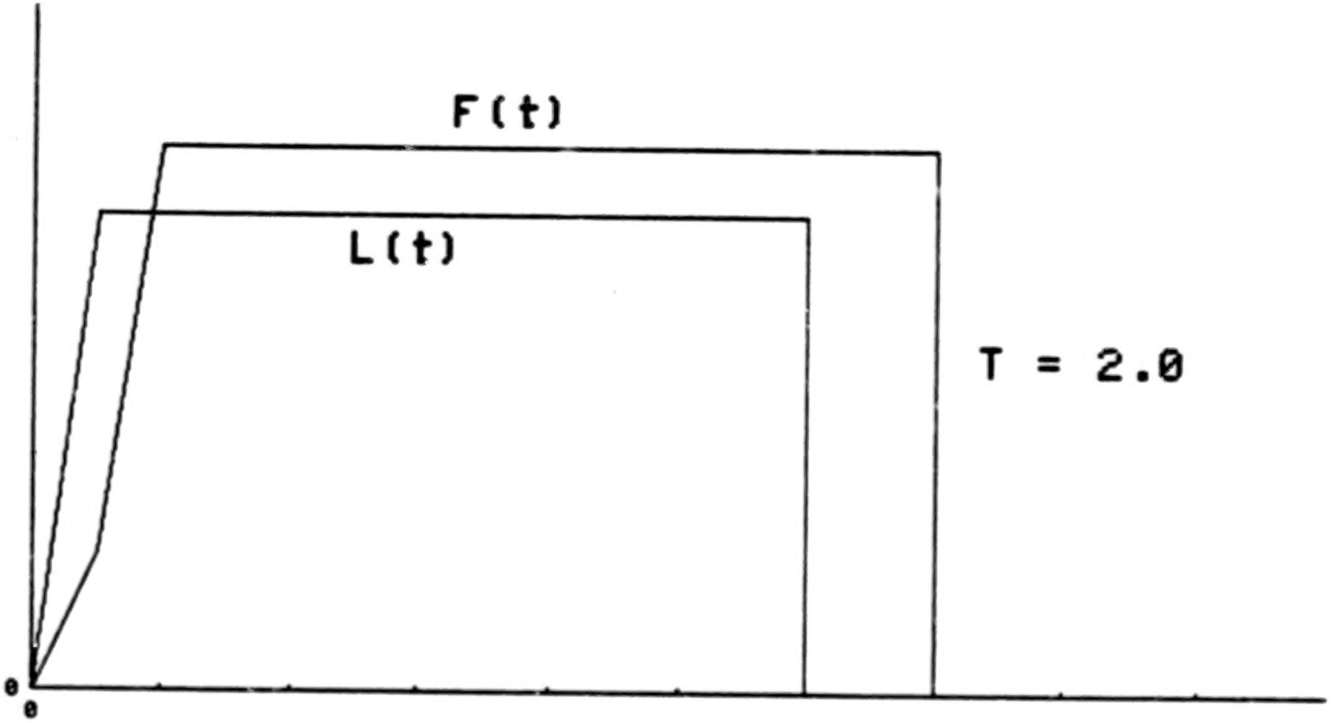

L(t)
BOX
L(t-T)
8
7
L(t)
L(t-T)
6
F(t)
F(t-T)
9
10
11
4
3
END
2
1
F(t)
F(t-T)

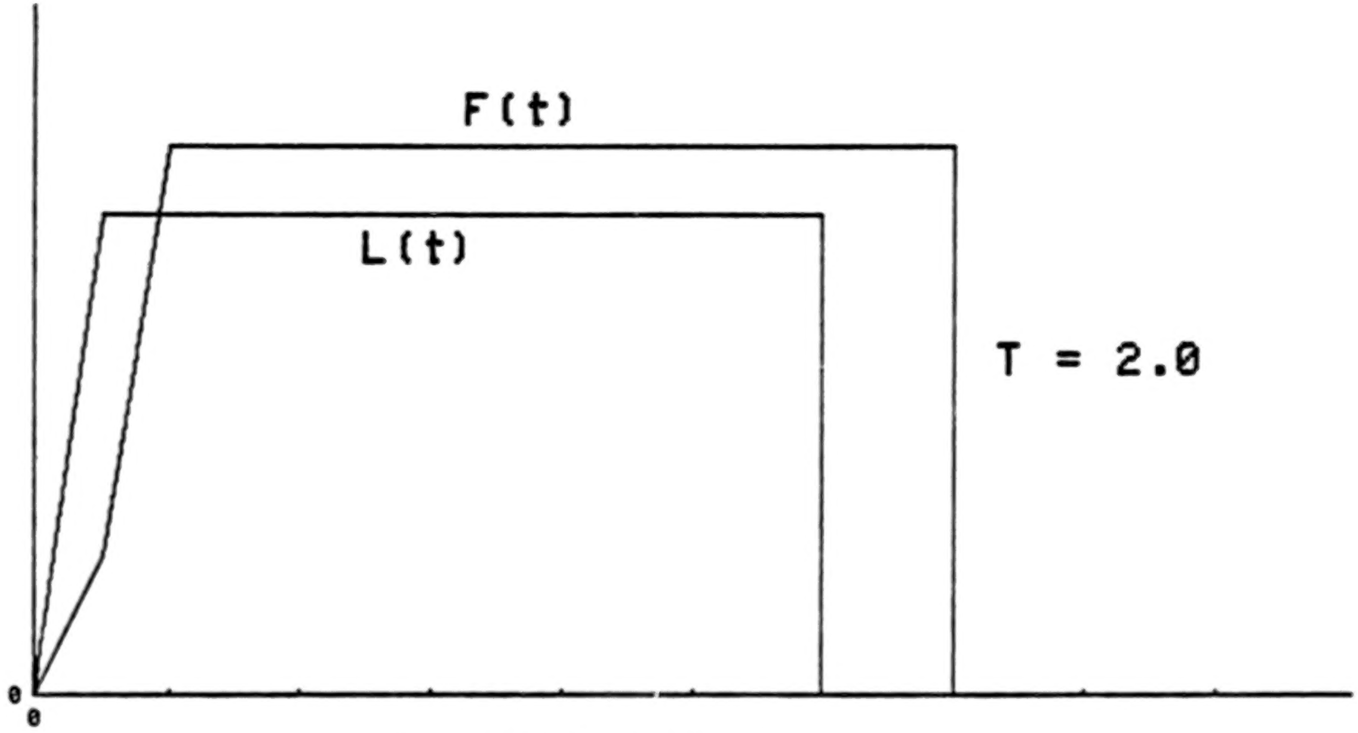
FIGURE 1

PILOT TEST STRUCTURE ALL

FIGURE 2



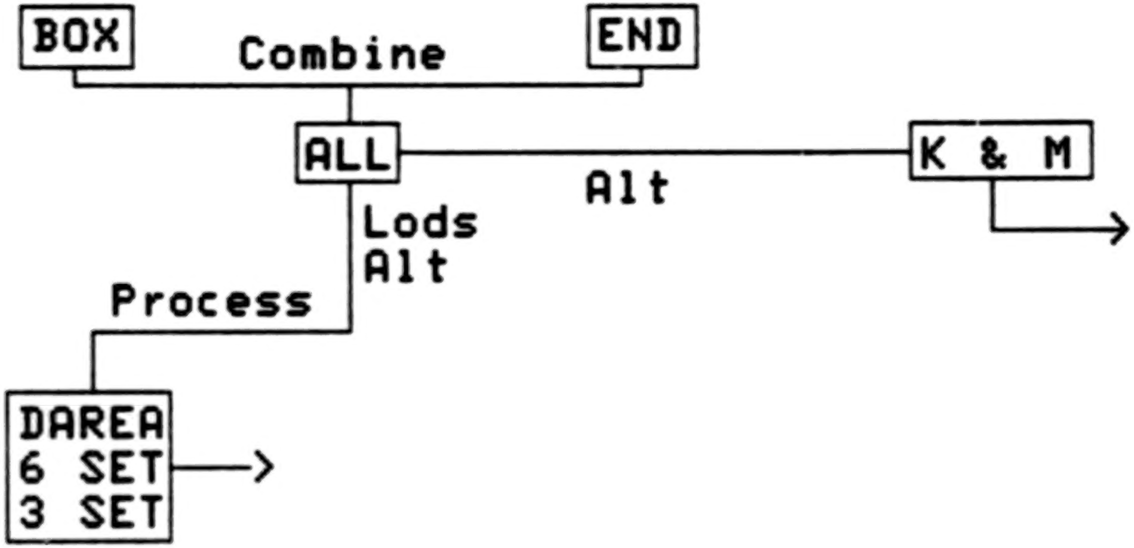
DYNAMIC LOADS



DYNAMIC LOADS

FIGURE 2

FIGURE 3


BOX
Combine
END
ALL
Alt
K & M
Lods
Alt
Process
DAREA
6 SET
3 SET

SUBSTRUCTURE TREE

S/S - R.F. 9

```
SCALARS     1 THRU 54

DAREA    2,4  LOAD
         5,7  LOAD
         COND  6-SET  &  8 LOAD
         COND  3-SET  &  1 LOAD

DLOAD,  TLOAD1,  TABLED1  (UNMOD)

DELAY     5,7 - SET   (UNMOD)
          COND  3-SET  &  1   TAU

ALT      INPUTT1  KMTX,  MMTX
         ADD     K4GG

OUTPUT     DISP - 11
           OLOAD - 9
```

FIGURE 4

SOLO

COMPLETE TRANSIENT

BULK OF ALL (BOX & END)

LOAD EXTERNAL BY GP & COMP

OMIT

OUTPUT DISP - 11
OLOAD - 9

FIGURE 5

DAREA
BEFORE OMIT
EXTERNAL SEQ.

G.P. 6

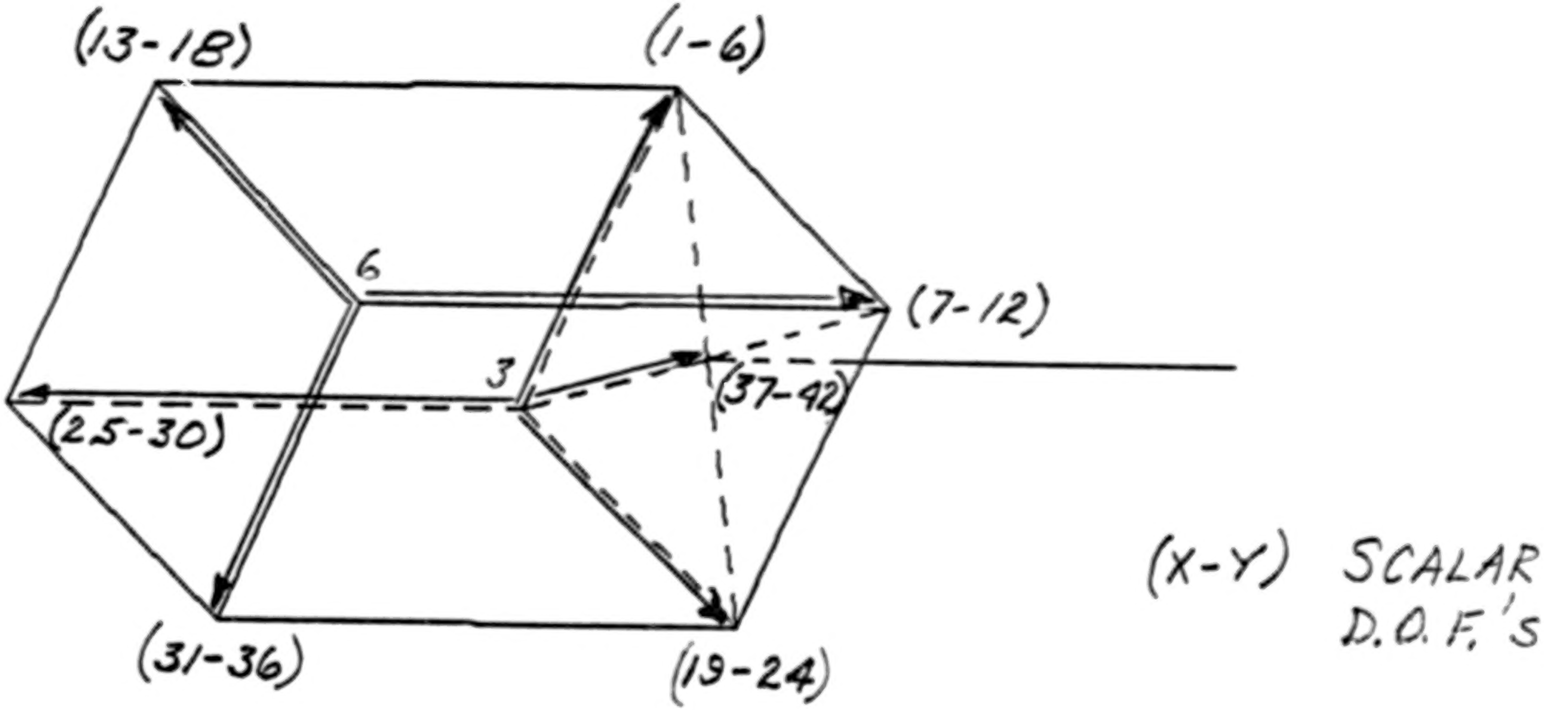
SID	GP	COMP	VALUE
68	6	1	1.0

G.P. 8

68	8	1	1.0

FIGURE 6

FIGURE 7


(13-18)
(1-6)
6
(7-12)
3
(37-42)
(25-30)
(31-36)
(19-24)
(X-Y) SCALAR
D.O.F.'S

REDISTRIBUTION
OF OMITTED LOADS

DAREA
AFTER OMIT
INTERNAL SEQ.

SID	GP	COMP	VALUE	GP	COMP	VALUE
68	7	0	$\boxed{.8}$	8	0	.02
68	9	0	-.02	10	0	.0
68	11	0	-.1	12	0	-.1
68	13	0	$\boxed{1.1}$	14	0	-.02
68	15	0	.002	16	0	-.01
68	17	0	.06	18	0	$\textcircled{.7}$
68	31	0	.1	32	0	-.002
68	33	0	.02	34	0	.01
68	35	0	$\textcircled{.7}$	36	0	.06

□ TRANSLATION ○ ROTATION

FIGURE 8

PHASE I ALTER
PACKET

```
ALTER   136
```

Just before END$

```
SOFI      /Pn,,,,/C,N,1/C,N,BOX/C,N,PVEC  $
```

Selects PL from SOF
Set n > largest S/S step #

```
UMERGE    USET,Pn,     /PLTOA/C,N,A/C,N,L/C,N,R  $
UMERGE    USET,PLTOA, /PATOF/C,N,F/C,N,A/C,N,O  $
UMERGE    USET,PATOF, /PFTON/C,N,N/C,N,F/C,N,S  $
UMERGE    USET,PFTON, /PGG  /C,N,G/C,N,N/C,N,M  $
```

Inflates PL to G-size with zeroes

```
CHKPNT    PGG $
SDR2      CASECC,CSTM,MPT,,EQEXIN,SIL,,,BGPDT,,,,,,PGG/
          OPG1,,,,,/C,N,STATICS/C,N,NOSORT2 $
```

Formats condensed load to GP's
according to Case Control

```
SAVE      NOSORT2
OFP       OPG1,,,,,//V,N,CARDNO $
```

Prepares output for Host computer

```
SAVE      CARDNO $
ENDALTER
```

TABLE I

PHASE II ALTER
PACKET

```
ALTER    136 Just before END $
SOFI        /KN,MN,PN,,/C,N,1/C,N,NAME/C,N,KMTX/C,N,MMTX/C,N,PVEC $
```

The Phase II load vector PVEC preserves the subcase ordering for all of Phase I DAREA cases. Set n ≥ max S/S step#

```
OUTPUT1   KN,MN,,,//C,N,-1/C,N,INP#/C,N,(CONDKM) $
```

K & M matrices of the final solution structure are output to disc for transfer to R.F. 9.

```
MATPRN    PN,,,,// $
```

Listing for man'l prep of DAREA input.

```
OUTPUT2   PN,,,,//C,N,-1/C,N,11/C,N,(LODOMIT) $
OUTPUT2,  ,,,,//C,N,-9/C,N,11 $
```

Matrix of condensed DAREA components is sent to FORTRAN logical disc file TAPE1 for prep of DAREA input by preprocessor.

```
ENDALTER
```

TABLE II

COMPARISONS OF DISPLACEMENT
RESPONSES AT END TIP (G.P.11)
BETWEEN COMPLETE R.F.9 & S/S R.F.9
AT EVERY SECOND FOR 7 SECONDS

T1		T2	
TRC	S/S	TRC	S/S
-4.000940+2	-4.000941+2	-3.4-5	-7.8-5
-7.335053+2	-7.335056+2	-2.2-4	-1.6-4
6.445970+2	6.445966+2	-5.4-4	-1.1-4
5.479066+3	5.479067+3	-7.3-4	+2.6-4
1.451452+4	1.451452+4	-9.9-6	+1.0-3
2.738419+4	2.738421+4	+3.2-4	+2.1-3
4.287670+4	4.287672+4	+9.2-5	+1.5-3

T3		R1	
TRC	S/S	TRC	S/S
3.159634+2	3.159632+2	-1.4-6	+1.5-5
5.792659+2	5.792651+2	-1.5-5	+6.5-5
3.949432+1	3.949254+1	-5.7-5	+1.6-4
-1.145368+3	-1.145371+3	-1.0-4	+3.2-4
-2.290734+3	-2.290735+3	-1.0-4	+5.4-4
-2.317057+3	-2.317051+3	+2.2-6	+7.6-4
-1.342825+3	-1.342805+3	+2.2-4	+9.6-4

TABLE III

COMPARISONS OF DISPLACEMENT RESPONSES AT END TIP (G.P.11) BETWEEN COMPLETE R.F.9 & S/S R.F.9 AT EVERY SECOND FOR 7 SECONDS

R2		R3	
TRC	S/S	TRC	S/S
-1.182428+1	-1.182427+1	-9.0-7	-1.0-6
-2.167783+1	_2.167781+1	-5.9-6	+2.4-6
-1.477996+0	-1.477947+0	-1.4-5	+1.7-5
4.286305+1	4.286311+1	-1.9-5	+4.5-5
8.572603+1	8.572604+1	+4.6-6	+8.0-5
8.671105+1	8.671079+1	+6.3-6	+1.0-4
5.025239+1	5.025163+1	-1.7-5	+3.0-5

TABLE IV

PHASE I STRATEGY
FOR DAREA CONDENSATION

EXECC SOL 2,0
ALTER

SUBSCC OPTIONS = K,M,P

CASECC OLOAD = ASET D.O.F.'s
S/C 1
LABEL = DAREA GP A COMP I VALUE R.R
LOAD = 1
S/C 2
LABEL = DAREA GP B COMP J VALUE S.S
LOAD = 2
.
.
S/C N
LABEL = DAREA GP M COMP K VALUE T.T

BULK FORCE1 . . . . GP A . . COMP I . . VALUE R.R
FORCE1 . . . . GP B . . COMP J . . VALUE S.S
FORCE1 . . . . GP M . . COMP K . . VALUE T.T

TABLE V

PHASE II STRATEGY
FOR DAREA CONDENSATION

EXECC	SOL 2,0 ALTER
SUBSCC	OPTIONS = K,M,P ANY COMBINES & REDUCES OUTPUT = SUMMARY OF PSEUDOSTRUCTURE CONNECTIVITIES
CASECC	DNA
BULK	DNA

TABLE VI

IMPLEMENTATION OF A TRAPEZOIDAL RING ELEMENT IN NASTRAN FOR ELASTIC-PLASTIC ANALYSIS

P. C. T. Chen and G. P. O'Hara
U. S. Army Armament Research and Development Command
Benet Weapons Laboratory, LCWSL
Watervliet Arsenal, Watervliet, NY 12189

SUMMARY

The explicit expressions for an elastic-plastic trapezoidal ring element are presented and implemented in NASTRAN computer program. The material is assumed to obey the von Mises' yield criterion, isotropic hardening rule and the Prandtl-Reuss flow relations. For the purpose of demonstration, two elastic-plastic problems are solved and compared with previous results. The first is a plane-strain tube under uniform internal pressure and the second, a finite-length tube loaded over part of its inner surface. A very good agreement has been found in both test problems.

INTRODUCTION

In recent years finite element method has been widely used for solving complex nonlinear problems and many large-scale general purpose computer programs have been developed (ref. 1). The MARC and ANSYS systems have found wide applications, yet they are quite expensive. The piecewise linear analysis option of the NASTRAN program provides an algorithm for solving nonlinear problems in material plasticity (ref. 2). The load is applied in increments such that the stiffness properties can be assumed to be constant over each increment. However, the usefulness of this option is quite limited because only a few elements have been implemented. These include rod, tube, bar elements for one-dimensional problems and plate elements for two-dimensional plane stress problems. This paper describes the implementation of a trapezoidal ring element in NASTRAN for solving elastic-plastic problems of rotational symmetry.

The theoretical basis of our implementation follows the approach first developed by Swedlow (ref. 3). A unique relationship between the octahedral stress and the plastic octahedral strain is assumed to exist. The material is assumed to obey the Mises' yield criterion, isotropic hardening rule and the Prandtl-Reuss flow relations. The explicit expressions for axisymmetric plasticity are derived. The element stiffness matrix and the stress data recovery routines for the trapezoidal ring element are developed. Seven new subroutines are implemented and added to the NASTRAN code.

For the purpose of demonstration, two elastic-plastic test problems are solved. The first is an infinitely long tube under uniform internal pressure. The NASTRAN results are in excellent agreement with an exact solution based on a finite-difference approach (ref. 4). The second problem is a thick-walled cylinder of finite length loaded over part of its inner surface. The NASTRAN results are compared with those obtained by a two-dimensional code with the use of quadrilateral ring elements (ref. 5). A good agreement between the two results has also been achieved.

CONSTITUTIVE RELATIONS

Following the development by Swedlow (ref. 1), the constitutive relations to be used in our formulation for solving elastic-plastic problems of rotational symmetry will be presented here. In the development, a unique relationship between the octahedral stress and the plastic octahedral strain is assumed to exist and the use of ideally plastic materials is excluded. The total strain components (ϵ_r, ϵ_θ, ϵ_z and γ_{rz}) are composed of the elastic, recoverable deformations and the plastic portions (ϵ_r^p, ϵ_θ^p, ϵ_z^p, and γ_{rz}^p). The rates of plastic flow, ($\dot\epsilon_r^p$, etc.), are independent of a time scale and are simply used for convenience instead of the incremental values. The definitions of the octahedral stress τ_o and the octahedral plastic strain rate $\dot\epsilon_o^p$ in the case of rotational symmetry are:

$$\tau_o = (1/3)[(\sigma_r-\sigma_\theta)^2 + (\sigma_\theta-\sigma_z)^2 + (\sigma_z-\sigma_r)^2 + 6\tau_{rz}{}^2]^{1/2} \tag{1}$$

$$\dot\epsilon_o^p = (1/3)[(\dot\epsilon_r^p-\dot\epsilon_\theta^p)^2 + (\dot\epsilon_\theta^p-\dot\epsilon_z^p)^2 + (\dot\epsilon_z^p-\dot\epsilon_r^p)^2 + (3/2)(\dot\gamma_{rz}^p)^2]^{1/2} \tag{2}$$

where (σ_r, σ_θ, σ_z, τ_{rz}) are the nonvanishing stress components.

A unique relationship between τ_o and ϵ_o^p is assumed and there exists a function, $M_T(\tau_o)$, such that

$$2M_T(\tau_o) = \dot\tau_o/\dot\epsilon_o^p \tag{3}$$

The plastic modulus, $M_T(\tau_o)$, can be related to the slope, E_T, of the effective stress-strain ($\bar\sigma$-$\bar\epsilon$) curve by

$$\frac{1}{3M_T(\tau_o)} = \frac{1}{E_T} - \frac{1}{E} \tag{4}$$

where E is the elastic modulus and

$$\begin{aligned} E_t &= \dot{\bar\sigma}/\dot{\bar\epsilon} \\ \bar\sigma &= (3/\sqrt{2})\tau_o \\ \bar\epsilon &= \sqrt{2}\,\epsilon_o + \bar\sigma/E\ . \end{aligned} \tag{5}$$

The material is assumed to obey the Mises yield criterion and the Prandtl-Reuss flow rule. The matrix relationship for the plastic flow in the case of rotational symmetry is

$$\begin{Bmatrix} \dot{\varepsilon}_r \\ \dot{\varepsilon}_\theta \\ \dot{\varepsilon}_z \\ \dot{\gamma}_{rz} \end{Bmatrix} = [C] \begin{Bmatrix} \dot{\sigma}_r \\ \dot{\sigma}_\theta \\ \dot{\sigma}_z \\ \dot{\tau}_{rz} \end{Bmatrix} \tag{6}$$

$$[C] = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 \\ -\nu & 1 & -\nu & 0 \\ -\nu & -\nu & 1 & 0 \\ 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} + \frac{1}{6\tau_o{}^2 M_T(\tau_o)} \begin{bmatrix} S_r{}^2 & S_r S_\theta & S_r S_z & 2S_r \tau_{rz} \\ & S_\theta{}^2 & S_\theta S_z & 2S_\theta \tau_{rz} \\ & & S_z{}^2 & 2S_z \tau_{rz} \\ \text{SYM} & & & 4\tau_{rz}{}^2 \end{bmatrix} \tag{7}$$

where ν is the Poisson's ratio,

$$S_r = (2\sigma_r - \sigma_\theta - \sigma_z)/3 \ ,$$
$$S_\theta = (2\sigma_\theta - \sigma_z - \sigma_r)/3 \ ,$$
$$S_z = (2\sigma_z - \sigma_r - \sigma_\theta)/3 \ . \tag{8}$$

For strain-hardening materials, M_T (or E_T) $\neq 0$, we can obtain the inverse of [C] numerically and this procedure is chosen in developing NASTRAN program. For ideally-plastic materials, $M_T = E_T = 0$, matrix [C] does not exist and the NASTRAN program fails. However, its inverse $[C]^{-1}$ still exists and the closed form has been derived in Ref. 6. In the axisymmetric case, the explicit form is (ref. 7)

$$[D] = \frac{2G}{1-2\nu} \begin{bmatrix} 1-\nu & & & \text{SYM} \\ \nu & 1-\nu & & \\ \nu & \nu & 1-\nu & \\ 0 & 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} - \frac{2G}{A} \begin{bmatrix} S_r{}^2 & & & \text{SYM} \\ S_r S_\theta & S_\theta{}^2 & & \\ S_r S_z & S_\theta S_z & S_z{}^2 & \\ S_r \tau_{rz} & S\ \tau_{tz} & S_z \tau_{rz} & \tau_{rz}{}^2 \end{bmatrix} \tag{9}$$

where

and

$$A = 3\tau_o^2(1 + M_T/G)$$

$$G = \frac{1}{2} E/(1+\nu) \ . \qquad (10)$$

If we want to remove the limitation that the use of ideally-plastic materials is excluded, we have to use Eq. (9) instead of Eq. (7).

TRAPEZOIDAL RING ELEMENT

The incremental displacement field employed for the trapezoidal ring element are

$$\Delta u(r,z) = \beta_1 + \beta_2 r + \beta_3 z + \beta_4 rz \ ,$$

$$\Delta w(r,z) = \beta_5 + \beta_6 r + \beta_7 z + \beta_8 rz \ . \qquad (11)$$

The transformation from grid point coordinates to generalized coordinates is

$$\{\beta\} = [\Gamma_{\beta q}]\{\Delta q\} \qquad (12)$$

where

$$\{\Delta q\}^T = [\Delta u_1, \Delta w_1, \Delta u_2, \Delta w_2, \Delta u_3, \Delta w_3, \Delta u_4, \Delta w_4] \ ,$$

$$\{\beta\}^T = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8] \ . \qquad (13)$$

and the elements of the inverse of the transformation matrix $[\Gamma_{\beta q}]^{-1}$ are the coefficients of the β's in Equations (11), evaluated at the corners of the element.

The stiffness matrix is formed in the same manner as that for the anisotropic elastic element. The final form referred to grid point coordinates is

$$[K] = [\Gamma_{\beta q}]^T[\bar{K}][\Gamma_{\beta q}] \ ,$$

where

$$[\bar{K}] = 2\pi\int r[B]^T[D][B]dzdr \ . \qquad (14)$$

[D], the matrix of material coefficients, is defined by Equation (9). The [B] matrix is the same as the elastic case, but now it expresses the incremental strains in terms of generalized coordinates

$$\{\Delta\varepsilon\} = [B]\{\beta\} \ . \qquad (15)$$

NASTRAN IMPLEMENTATION

The NASTRAN implementation for an elastic-plastic trapezoidal ring element follows the steps outlined in section 6.8, "Adding a structural element," of reference 8. Changes were required in the functional modules PLA1, PLA3, and PLA4, which included the writing of several new subroutines. These new routines could easily be modeled after the existing code for the linear portion of the program. There are two major differences in the nonlinear subroutines. First is the new code for the calculation of the material stiffness matrix and second is that thermal stresses and element force calculations are eliminated in the nonlinear code.

The changes in PLA1 allows this module to identify the new element as a member of the piecewise linear element set and properly initialize the nonlinear Element Summary and Element Connection Property Tables. Three element stress recovery subroutines were added to PLA3: PSAPRG, a driver for stress data recovery; PSTRR1 and PSTRR2, phase I and II stress recovery routines. Element stiffness calculations in PLA4 require four new subroutines: PKAPRG, a driver for nonlinear trapezoidal ring elements in PLA4, PKTRR1 and PKTRR2, stress recovery routines which generate stresses for the computation of the nonlinear material matrix; and PKTRAP, the stiffness matrix generation routine for nonlinear trapezoidal ring elements.

The computer system available for this work is IBM 360 Model 44. In order to add the new code into Link 13, it became necessary to add a new branch to the overlay trees to contain the new elements.

NUMERICAL EXAMPLES

For the purpose of demonstration, two elastic-plastic problems of rotational symmetry were solved and compared with other results (refs. 4 and 5). The first is an infinitely long tube under uniform internal pressure and the second, a thick-walled cylinder of finite length loaded over part of its inner surface. The material properties for both problems are the same. The elastic constants are $E = 30\mathrm{x}10^6$ psi, $\nu = 0.3$ and the effective stress-strain curve is represented by three line segments connecting the four points in the $(\bar{\varepsilon}-\bar{\sigma})$ plane, $(\bar{\varepsilon},\bar{\sigma}) = (0.0,0.0)$. (0.005, 150,000 psi), (0.055, 225,000 psi), (0.1, 225,000 psi).

EXAMPLE 1. Consider an infinitely long tube subjected to uniform internal pressure p. The plane strain condition is assumed. The tube of outside radius 2" and inside radius 1" has been divided into 25 trapezoidal ring elements. The numerical results based on the NASTRAN program have been obtained. For this problem, a new finite-difference approach (ref. 4) can be used to generate exact solution and to assess the accuracy of the NASTRAN code. Some of the results for the displacements and stresses are presented graphically in Figures 1 and 2. Twenty-five load increments are used in NASTRAN as shown in Figure 1. The radial displacements at the inside as well as outside surface are shown as functions of internal pressure. Figure 2 shows the distributions of radial, tangential and axial stress components in a pressurized tube when half of the tube is plastic. The pressure required to achieve this state is 0.7378 σ_0 based on NASTRAN code and 0.7356 σ_0 based on the finite-difference solution (ref. 4). Both codes indicate that the maximum tangential stress occurs at the elastic-plastic interface. As demonstrated in Figures 1 and 2, the NASTRAN results are in excellent agreement with those based on the finite-difference approach (ref. 4).

EXAMPLE 2. Consider a two-dimensional elastic-plastic thick-walled cylinder problem as shown in Figure 3. The tube with inner radius (1"), outer radius (2") and length (4") is loaded uniformly over a middle portion (2") of the inner surface. The mesh generation and the loading for the half of the undeformed structure is shown in the figure. This problem was solved in (ref. 5) based on a scale loading approach. The first load factor is the upper limit of the elastic solution and ten additional increments were needed until one of the outside elements becomes yielded. The same load factors were used to obtain the NASTRAN solution. Both programs indicate that the sequence in which the elements become plastic is 1,5,9,2,13,6,10,3,7,14,11,17,4. Some steps will cause more than one element to become plastic and those elements with effective stress $\bar{\sigma} \geq 0.99\ \bar{\sigma}_0$ have been considered as yielded. The numerical results for the radial displacement at the inside, u_a (point 1) and outside, u_b (point 5) as functions of internal pressure are shown in Figure 4. The stress components at the centroid of one inside element (No. 1) are shown in Figure 5. The effect of loading history on the displacements and stresses can be seen from Figures 4 and 5. A comparison of the results between NASTRAN program and reference 5 indicates that very good agreement has been achieved.

CONCLUSION

An elastic-plastic trapezoidal ring element has been implemented in NASTRAN computer program. Its application to elastic-plastic problems of rotational symmetry has been demonstrated by solving two thick-walled tube problems. The NASTRAN results for both problems are in excellent agreement with the other results.

REFERENCES

1. W. Pilkey, K. Saczalski, and H. Schaeffer, "Structural Mechanics Computer Programs, Surveys, Assessments and Availability," University Press of Virginia, 1974.

2. "NASTRAN Theoretical Manual," NASA SP-221 (01), 1972.

3. J. L. Swedlow, "The Thickness Effect and Plastic Flow in Cracked Plates," ARL 65-216, Aerospace Research Laboratory, Office of Aerospace Research, October 1965.

4. P. C. T. Chen, "A Finite-Difference Approach to Axisymmetric Plane Strain Problems Beyond the Elastic Limit," To appear in Trans. 25th Conference of Army Mathematicians, 1979.

5. P. C. T. Chen, "Numerical Solution of Gun Tube Problems in the Elastic-Plastic Range," Proceedings of the 1977 Army Numerical Analysis and Computer Conference, pp. 423-439.

6. Y. Yamada, N. Yoshimura, and T. Sakurai, "Plastic Stress-Strain Matrix and Its Application for the Solution of Elastic-Plastic Problems by the Finite Element Method," Int. J. Mech. Sci., Vol. 10, pp. 345-354, 1968.

7. P. C. T. Chen, "Elastic-Plastic Solution of a Two-Dimensional Tube Problem by the Finite Element," Trans. Nineteenth Conference of Army Mathematicians, pp. 763-784.

8. "NASTRAN Programmers' Manual," NASA SP-223 (01), 1972.

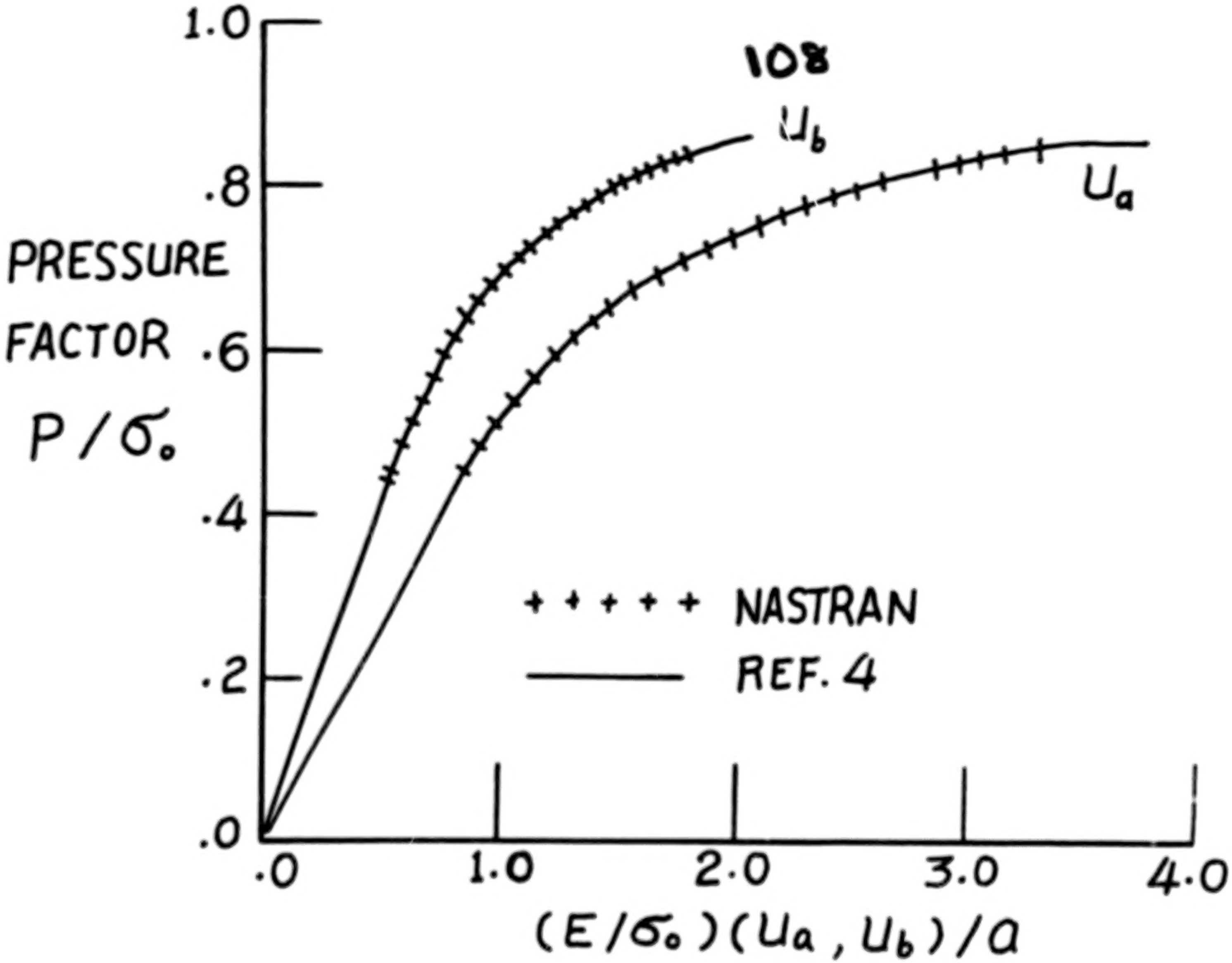

1.0
108
U_b
.8
U_a
PRESSURE FACTOR
P/σ_0
.6
.4
+ + + + + NASTRAN
REF. 4
.2
.0
.0
1.0
2.0
3.0
4.0
$(E/\sigma_0)(U_a, U_b)/a$

FIGURE -1

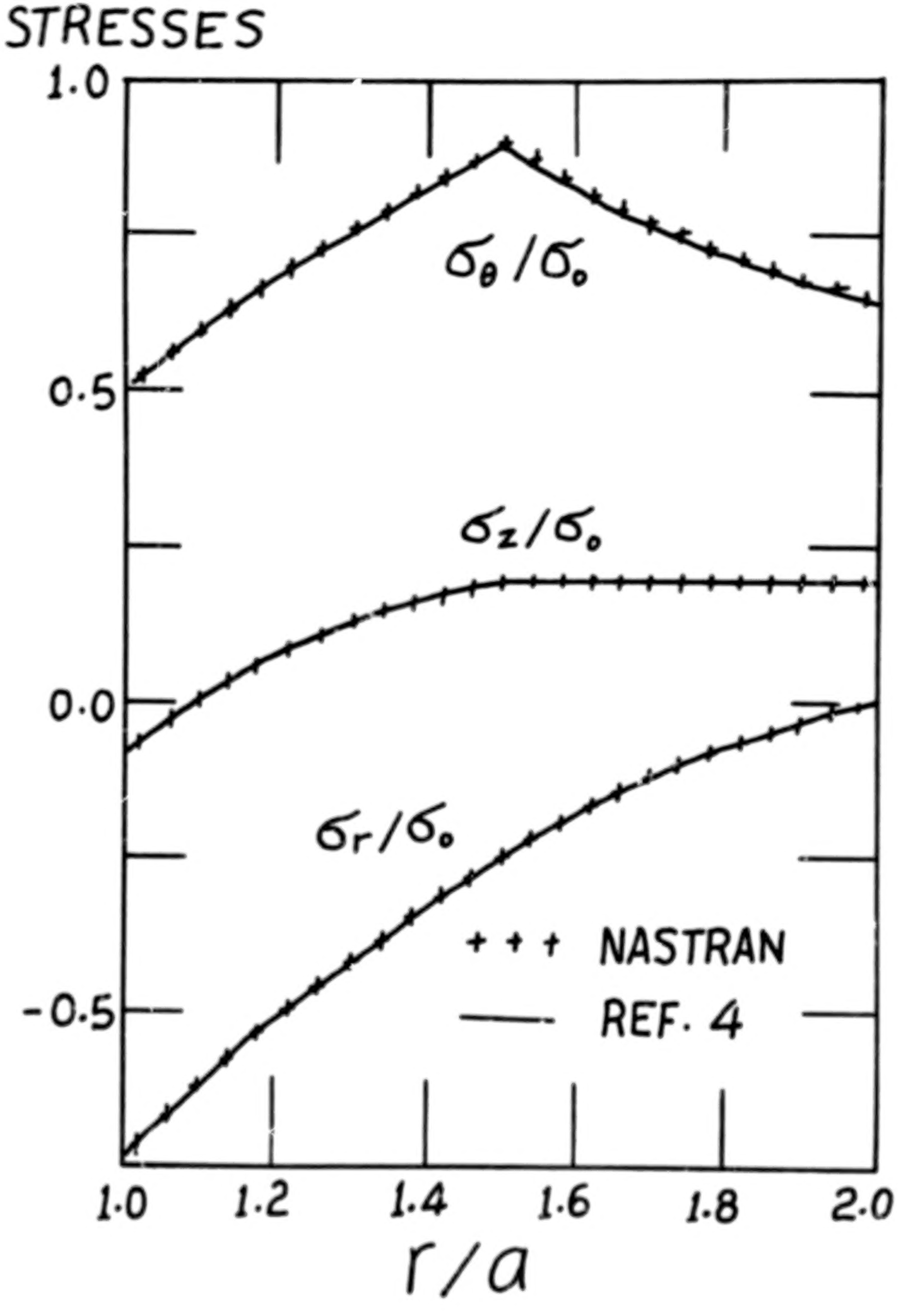

STRESSES
1.0
0.5
0.0
-0.5
σ_θ/σ_0
σ_z/σ_0
σ_r/σ_0
+ + + NASTRAN
REF. 4
1.0
1.2
1.4
1.6
1.8
2.0
r/a

FIGURE - 2

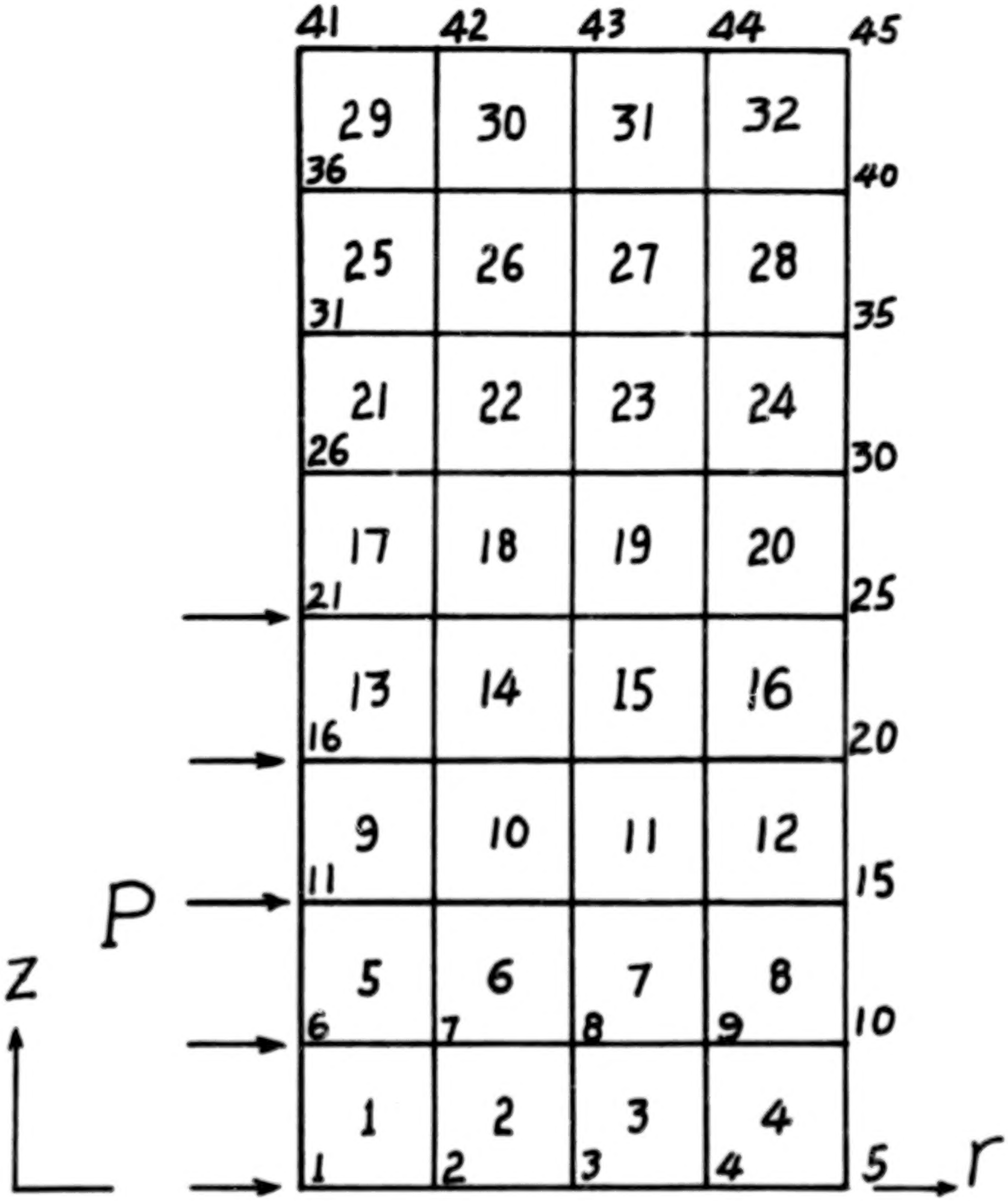

P
Z
r

FIGURE - 3

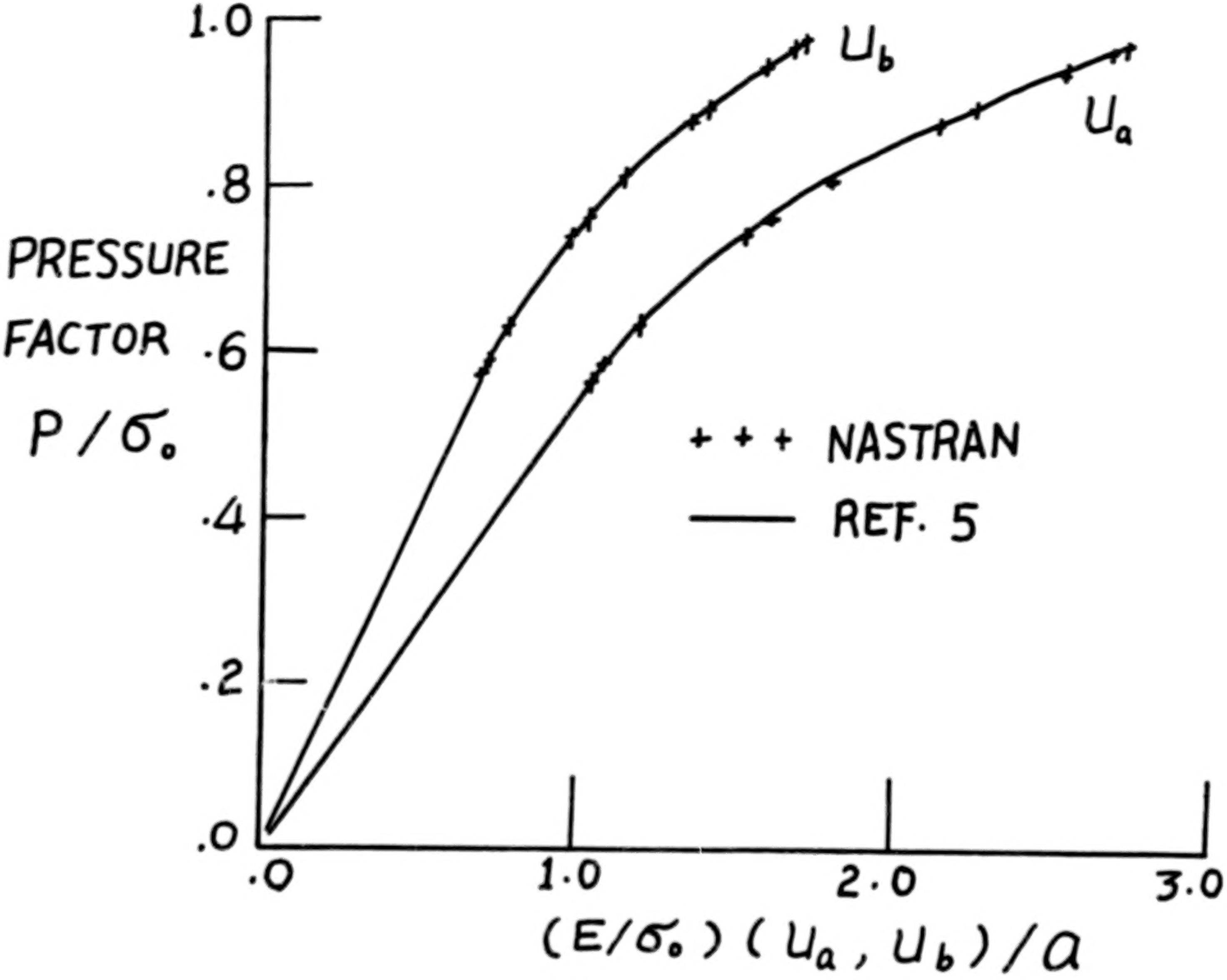

1.0
.8
.6
.4
.2
.0
PRESSURE
FACTOR
P / σ₀
.0
1.0
2.0
3.0
(E/σ₀)(Ua, Ub)/a
Ub
Ua
+ + + NASTRAN
REF. 5

FIGURE - 4

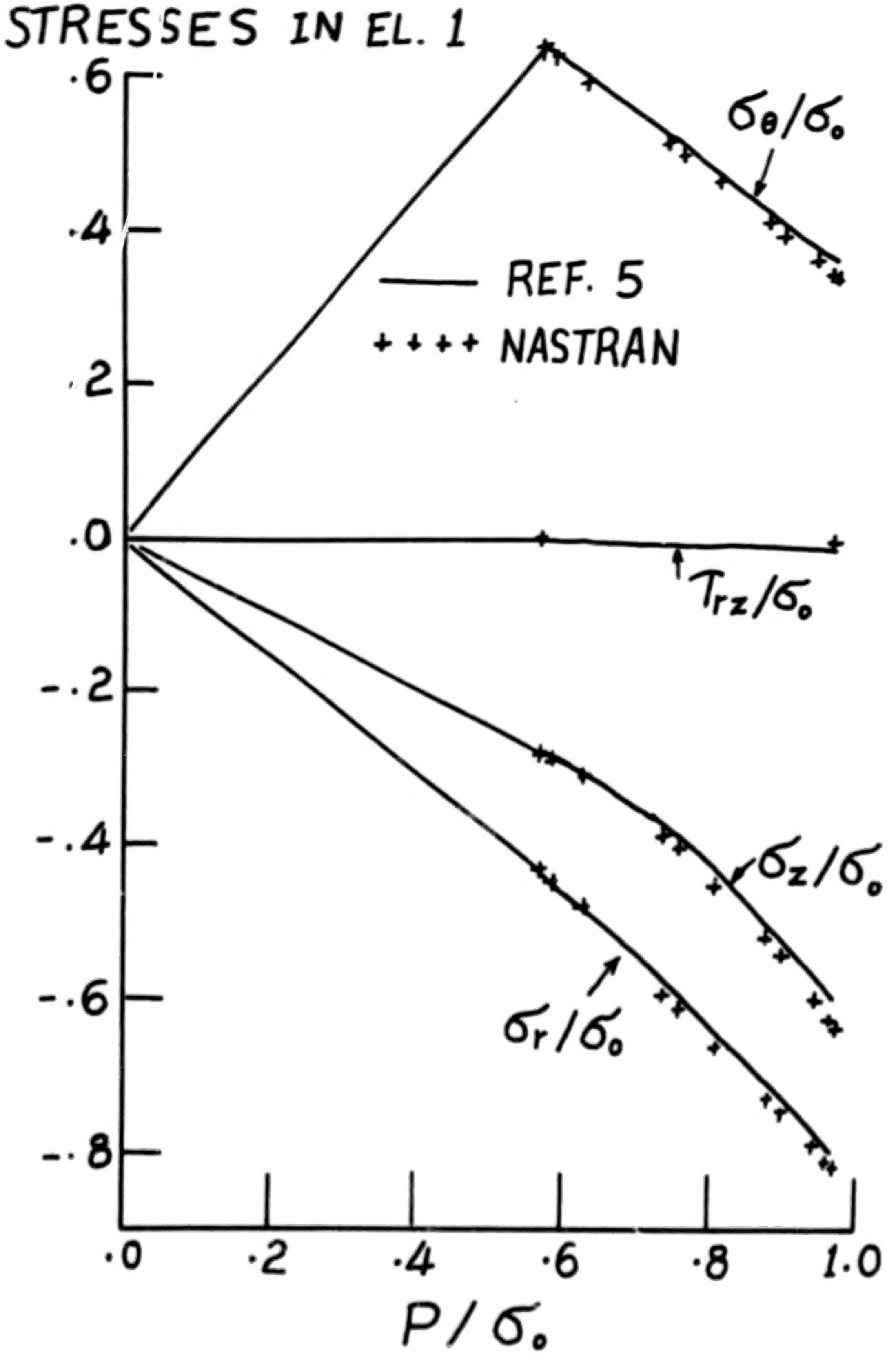

STRESSES IN EL. 1
·6
·4
·2
·0
-·2
-·4
-·6
-·8
·0
·2
·4
·6
·8
1.0
P/σ_o
σ_θ/σ_o
REF. 5
+ + + + NASTRAN
τ_{rz}/σ_o
σ_z/σ_o
σ_r/σ_o

FIGURE - 5

NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT*

Karl A. Meyer
Planning Research Corporation
John F. Kennedy Space Center, Florida

SUMMARY

Special modeling techniques were developed to simulate an accurate mathematical model of the Transporter/Canister/Payload (T/C/P) system during ground transport of the Inertial Upper Stage/Tracking and Data Relay Satellite (IUS/TDRS) payload on the John F. Kennedy Space Center (KSC). The three finite element models -- the transporter, the canister, and the IUS/TDRS payload -- were merged into one model and used along with the NASTRAN normal mode analysis. Deficiencies were found in the NASTRAN program that make a total analysis using modal transient response impractical. It was also discovered that inaccuracies may exist for NASTRAN rigid body modes on large models when Given's method for eigenvalue extraction is employed. The deficiencies as well as recommendations for improving the NASTRAN program are discussed.

INTRODUCTION

The capability to predict the operational life of a Space Shuttle payload is essential to the success of an orbital mission. The amplitudes and approximate number of load cycles that will be induced in each individual payload during ground transport between facilities must be predictable in advance of each flight into space.

This paper presents a solution to the problem of determining the dynamic response of one particular payload: the IUS/TDRS. The TDRS, which will be boosted to a geo-synchronous Earth orbit by the IUS, presents a problem in that it has a flexible diaphragm tank that contains approximately 1,400 pounds of hydrazine fuel. The tank is extremely sensitive to cyclic fatigue.

Though the original purpose of the study was to determine the dynamic response of the IUS/TDRS payload to ground-induced, time-dependent displacements

---

* This work was done under NASA contract No. NAS10-8525

using the NASTRAN computer program, it became apparent that a modal transient response analysis (solution 12) was not feasible due to deficiencies in the program.

Instead, it was necessary to perform a normal mode analysis (solution 3). Selected results from the NASTRAN program were used as input to a FORTRAN program in which time-dependent displacements were imposed and the response was computed.

THE T/C/P SYSTEM

The T/C/P system transports Space Shuttle payloads between facilities at KSC. It consists of a canister that houses the payload and a transporter on which the payload canister is carried. The canister is a steel shell with rigid steel frames and two large aluminum doors; it can be used in either a vertical or horizontal position. The transporter has a steel flatbed frame that contains supporting subsystems such as environmental control, instrumentation and communication, fluids and gases, as well as the diesel generators required for self-propulsion. The frame is mounted on 12 bogie units (6 drive units and 6 braking units) with 4 tires per unit.

The IUS/TDRS payload, the subject of this study, is carried with the canister in the vertical position.

NORMAL MODE ANALYSIS

To perform a normal mode analysis, a NASTRAN model was developed to determine the normal mode shapes, eigenvalues, and generalized mass for selected mode shapes as well as the total mass, mass moment of inertia, and center-of-gravity location of the T/C/P system. It contained 900 grid points in which ASET1 (ref. 1) bulk data cards were used to reduce the mass Degrees of Freedom to a total of 338, using the Guyan reduction technique and the Givens method for eigenvalue extraction. A total of 2915 CBAR, CELAS2, CQUAD1, CTRIA1, and CONM2 elements were used in the model.

Figure 1 shows a NASTRAN undeformed structural plot of the T/C/P system. Figure 2 shows a NASTRAN undeformed structural plot of the T/C/P system in which the canister was plotted using NASTRAN PLOTEL elements. The second plot configuration was used for modal deformed plots to clarify mode shapes.

Modeling - IUS/TDRS Payload

The IUS/TDRS payload was modeled using CBAR and CØNM2 elements and connected to the payload canister with CELAS2 elements. The IUS/TDRS payload was modeled in its own coordinate system using CØRD2R bulk data cards.

Modeling - Payload Canister

The payload canister was modeled using CBAR, CQUAD1, CTRIA1, and CØNM2 elements. Connections between payload canister and transporter were made using CELAS2 elements. The payload canister steel shell was in the basic coordinate system, and each canister aluminum door was modeled in its own coordinate system using CØRD2R bulk data cards.

Modeling - Transporter

The transporter was modeled using CBAR, CTRIA1, and CØNM2 elements. It was modeled in its own coordinate system using the CORD2R bulk data cards.

DISCUSSION

Each component of the T/C/P system was modeled in its own coordinate system for ease of analysis for future payloads. This modeling technique is particularly useful for cases in which the location of the payload relative to the payload canister may change and/or the transportation position of the payload canister could be either horizontal or vertical. It points out the value of the NASTRAN coordinate system bulk data cards: components of a total structural system may be translated and/or rotated relative to one another, and only minor changes in the NASTRAN bulk data cards are required.

The NASTRAN PARAM GRDPNT bulk data card was used to locate center of gravity, to compute total mass, and to compute mass moment of inertia relative to the center of gravity. This output was used as input to the previously mentioned FORTRAN program.

When modeling a plane frame in which the members have open sections and substantial depth and are rigidly connected to one another, as in the transporter bed frame, the torsional mode eigenvalues are inaccurate unless a specific modeling technique is used. This inaccuracy occurs because warping normal and warping shear strains (ref. 2) resulting from internal torsional loads are not properly accounted for in the model and result in torsional mode eigenvalues that are much smaller in value than actually occur.

To test this fact, two small NASTRAN models of plane frames were analyzed. Both models consisted of plane frames made up of I-shaped members; flanges of the members were parallel to the plane of the frame. NASTRAN CBAR elements were used in both models. The second model had additional members that consisted of CBAR elements offset from the neutral axis of the total section to the neutral axis of each flange. The PBAR bulk data card for these offset CBAR elements consisted only of moment of inertia in the plane of the flange. All other values of this PBAR bulk data card were zero. This modeling technique produces additional stiffness terms that increase the torsional rigidity of the structural elements. A NASTRAN undeformed structural plot of these models is shown in figure 3.

A normal mode analysis was performed on both models. The results of the first model analysis gave a first torsional mode frequency of 2.716209 Hz. The second model first torsional mode frequency was 13.55733 Hz. The error of the first model torsional frequency relative to the second model torsional frequency was 499%. Figures 4 and 5 show modal deformed plots of the first torsional modes of the first and second models, respectively.

The first model and second model first bending mode frequencies were 44.00233 Hz and 44.42864 Hz, respectively, which shows that the modeling technique has little effect on pure bending mode. It also points out the importance of the offset feature on the NASTRAN CBAR bulk data card. Figures 6 and 7 show modal deformed plots of the first and second model, respectively.

This modeling technique was used on the transporter bed frame. Tests were run on a similar type frame and results show that this and similar modeling techniques produce accurate eigenvalues.

NASTRAN RESULTS

The T/C/P system was analyzed using the Givens method for eigenvalue extraction in a free body configuration. A Univac 1108 computer was used with level 16.0 NASTRAN.

Table I summarizes the T/C/P system real eigenvalues and frequencies for the first six rigid body modes. The eigenvalues and frequencies are not zero, thus raising the question of the accuracy of these modes. Also, the force and stress output of NASTRAN showed appreciable strain energy at certain locations, which raised additional doubts as to the the accuracy of the rigid body modes. Because of these questions, the mode shapes were hand calculated at points of interest and used as input to the previously mentioned FORTRAN program.

Table II summarizes the T/C/P system real eigenvalues, frequencies, and mode type for the first 12 flexible body modes. No tests related to this T/C/P system have been made at this time, so no comment can be made on the accuracy of these modes. Figures 8 and 9 show typical modal deformed plots of the T/C/P system.

It should also be noted that the Forward - Backward Substitution time in module SMP1 was approximately 23,000 CPU-seconds on a Univac 1108 computer running level 16.0 NASTRAN.

DEFICIENCIES IN NASTRAN

It would have been desirable to run a full analysis of the T/C/P system (solution 12, Modal Transient Response); however, certain deficiencies became apparent as the analysis progressed. These include:

1. Questionable accuracy of rigid body modes when using Givens method for eigenvalue extraction for free body systems
2. No provisions for non-zero initial conditions relative to modal transient response rigid format
3. No provisions for time-dependent or frequency-dependent displacements
4. No provisions for checking accuracy of numerical integration for each time or frequency step

Alteration of the NASTRAN program and the use of special modeling techniques were considered; however, the time frame of the analysis and the magnitude of these deficiencies made these considerations infeasible.

NASTRAN PROGRAM RECOMMENDATIONS

The following recommendations are made for the improvement of the NASTRAN program for dynamic modal response problems:

1. Develop a timing equation check of the Forward - Backward Substitution method used by NASTRAN.
2. Check the accuracy of rigid body modes for large problems for all types of eigenvalue extraction methods.
3. Make provisions for non-zero initial conditions as NASTRAN bulk data input in the solution of modal response type of rigid formats.
4. Add time and frequency-dependent displacement equations in the form of NASTRAN bulk data cards.
5. Develop a numerical integration technique (variable step integration) that would check each integration step for accuracy. Some allowable maximum error should be used as input by the user in the form of a NASTRAN bulk data parameter card.

CONCLUSIONS

NASTRAN is an excellent tool when used for the extraction of eigenvalues and eigenvectors. The use of several methods of eigenvalue extraction and the existence of useful types of NASTRAN bulk data cards, such as ASET1, coordinate system cards, and PARAM GRDPNT bulk data cards, make it an efficient, time-saving program.

The author concludes, however, that considerable care must be exercised in the choice of modeling techniques and that there are deficiencies in the NASTRAN program that make a total analysis of some types of problems using modal transient response impractical. The above-mentioned recommendations would improve the NASTRAN program, making it more useful and efficient for solving modal response-type problems.

Table I. NASTRAN Rigid Body Modes

MODE NO.	EIGENVALUE	FREQUENCY (HZ)
1	−5.583629	0.376078
2	−3.456232	0.295884
3	−0.273054	0.083165
4	1.667725	0.205533
5	4.727485	0.346047
6	5.503941	0.373385

Table II. NASTRAN Flexible Body Modes

MODE NO.	EIGENVALUE	FREQUENCY (HZ)	MODE TYPE
7	723.2128	4.280093	IUS/TDRS BENDING
8	812.9676	4.537919	IUS/TDRS BENDING
9	1140.882	5.375767	T/C/P SYSTEM COUPLED
10	1478.756	6.120238	TRANSPORTER BENDING
11	2033.65	7.177253	TRANSPORTER TORSIONAL
12	2330.44	7.683147	IUS/TDRS BENDING
13	2685.299	8.247388	IUS/TDRS BENDING
14	2982.683	8.692079	IUS/TDRS BENDING
15	3236.141	9.053861	T/C/P COUPLED
16	3517.725	9.439545	PAYLOAD CANISTER BENDING
17	3751.85	9.748613	T/C/P COUPLED
18	3880.339	9.914138	T/C/P COUPLED

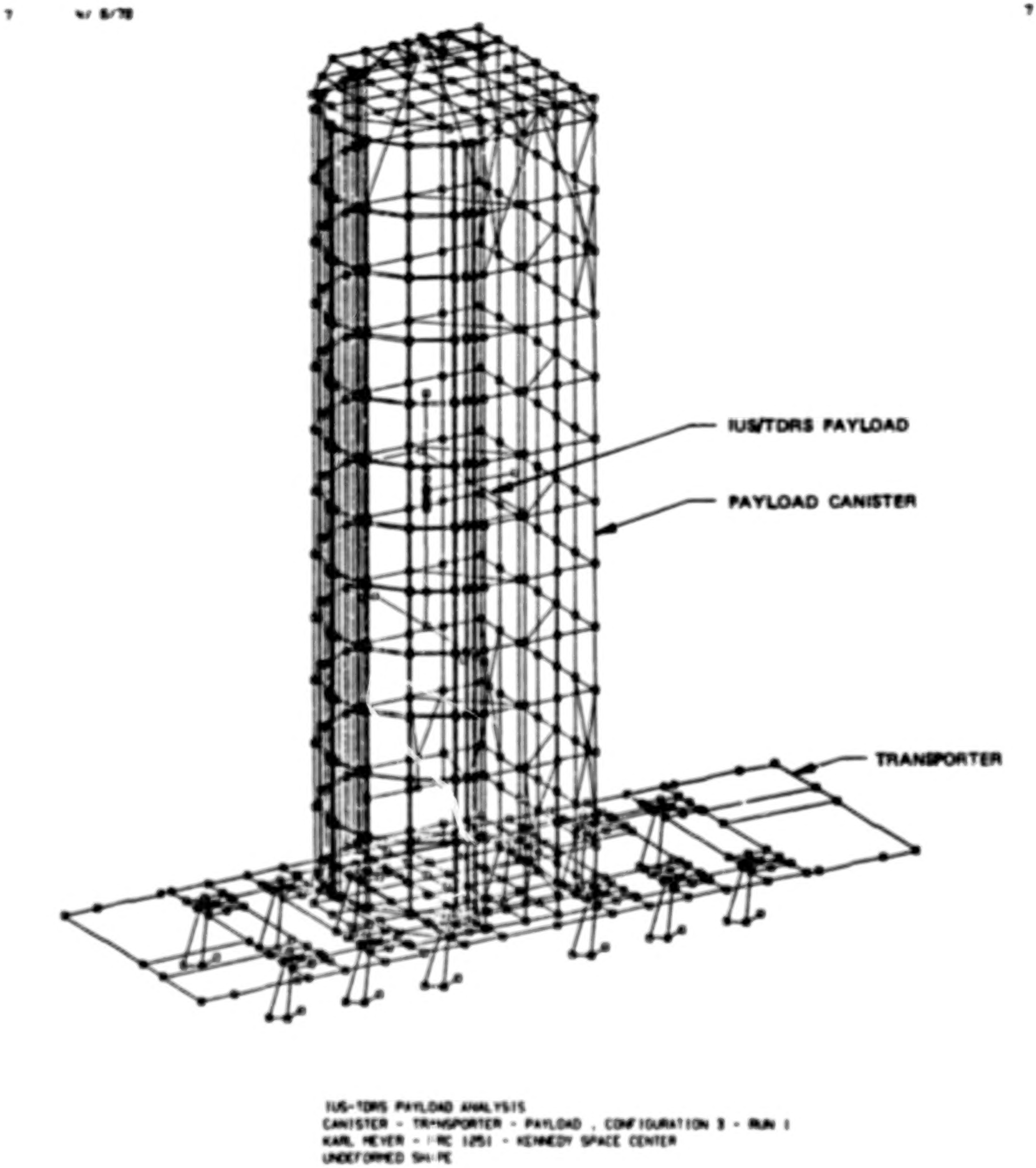

7
7
IUS/TDRS PAYLOAD
PAYLOAD CANISTER
TRANSPORTER
IUS-TDRS PAYLOAD ANALYSIS
CANISTER - TRANSPORTER - PAYLOAD , CONFIGURATION 3 - RUN 1
KARL MEYER - FRC 1251 - KENNEDY SPACE CENTER
UNDEFORMED SHAPE

Figure 1. NASTRAN Undeformed Structural Plot of the T/C/P System

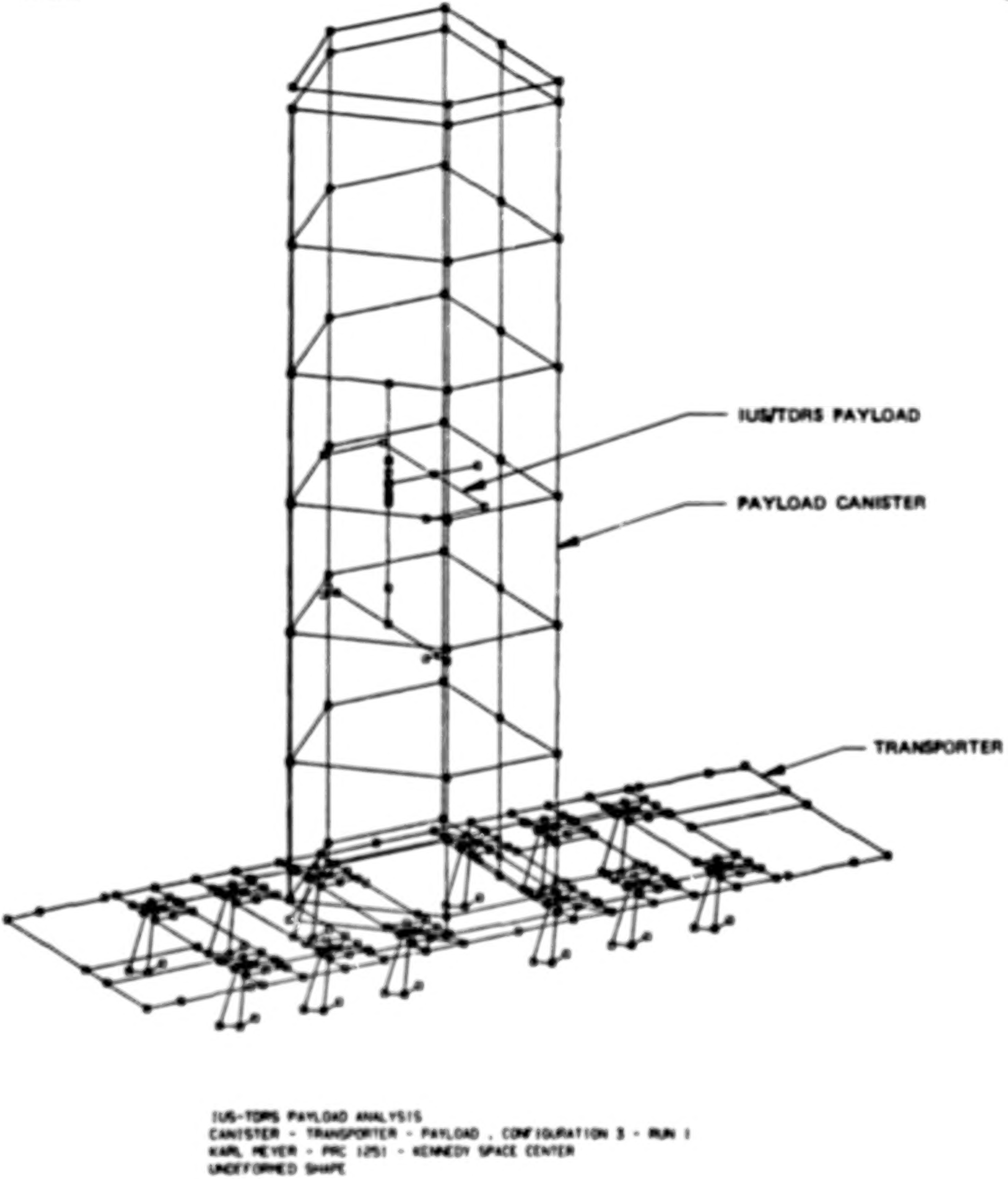

IUS/TDRS PAYLOAD
PAYLOAD CANISTER
TRANSPORTER
IUS-TDRS PAYLOAD ANALYSIS
CANISTER - TRANSPORTER - PAYLOAD , CONFIGURATION 3 - RUN 1
KARL MEYER - PRC 1251 - KENNEDY SPACE CENTER
UNDEFORMED SHAPE

Figure 2. NASTRAN Undeformed Structural Plot of T/C/P System with PLOTEL used for Canister Plot

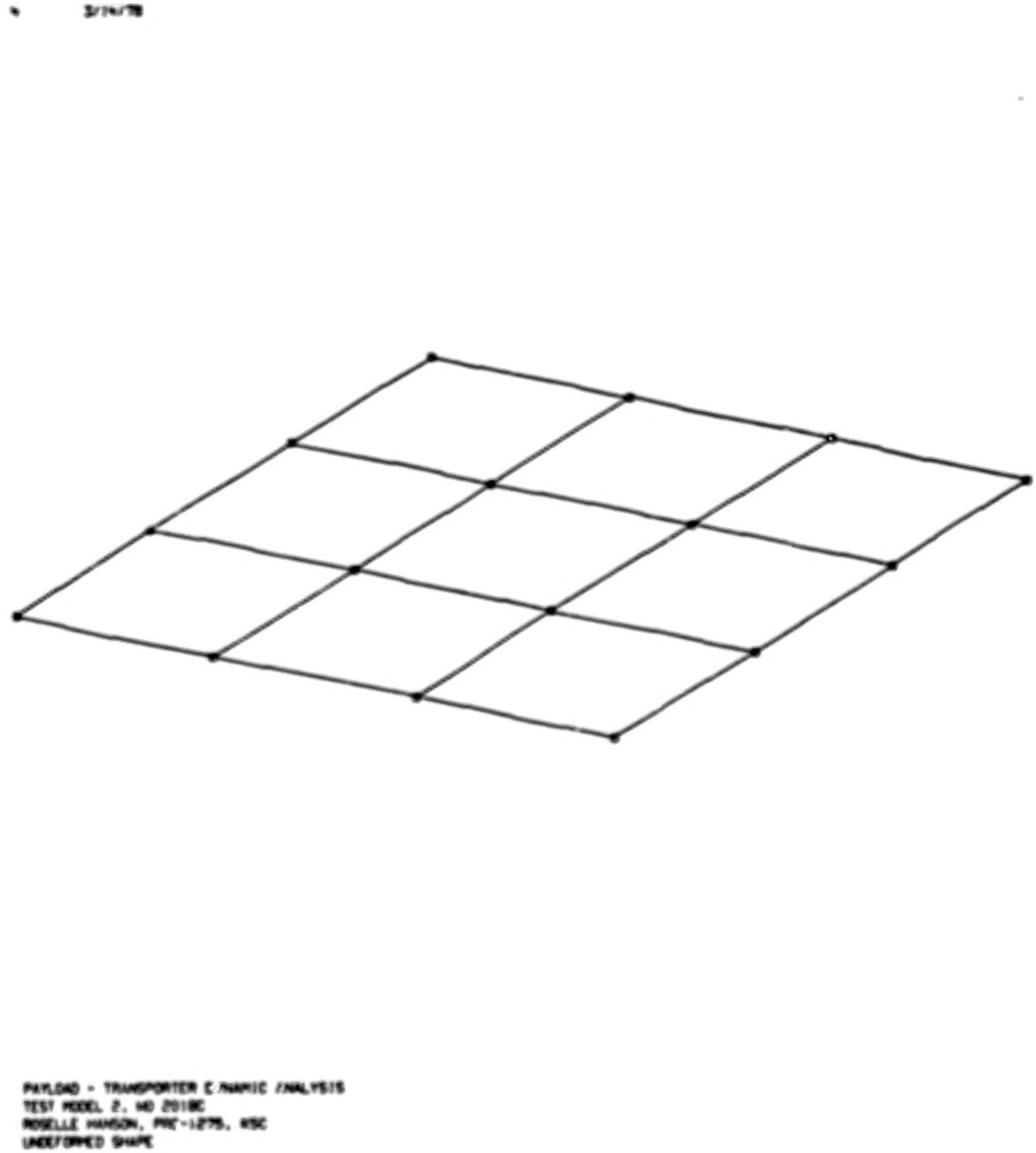

3/14/78
PAYLOAD - TRANSPORTER DYNAMIC ANALYSIS
TEST MODEL 2, HD 2018C
ROSELLE HANSON, PRC-1275, KSC
UNDEFORMED SHAPE

Figure 3. Test Models 1 and 2

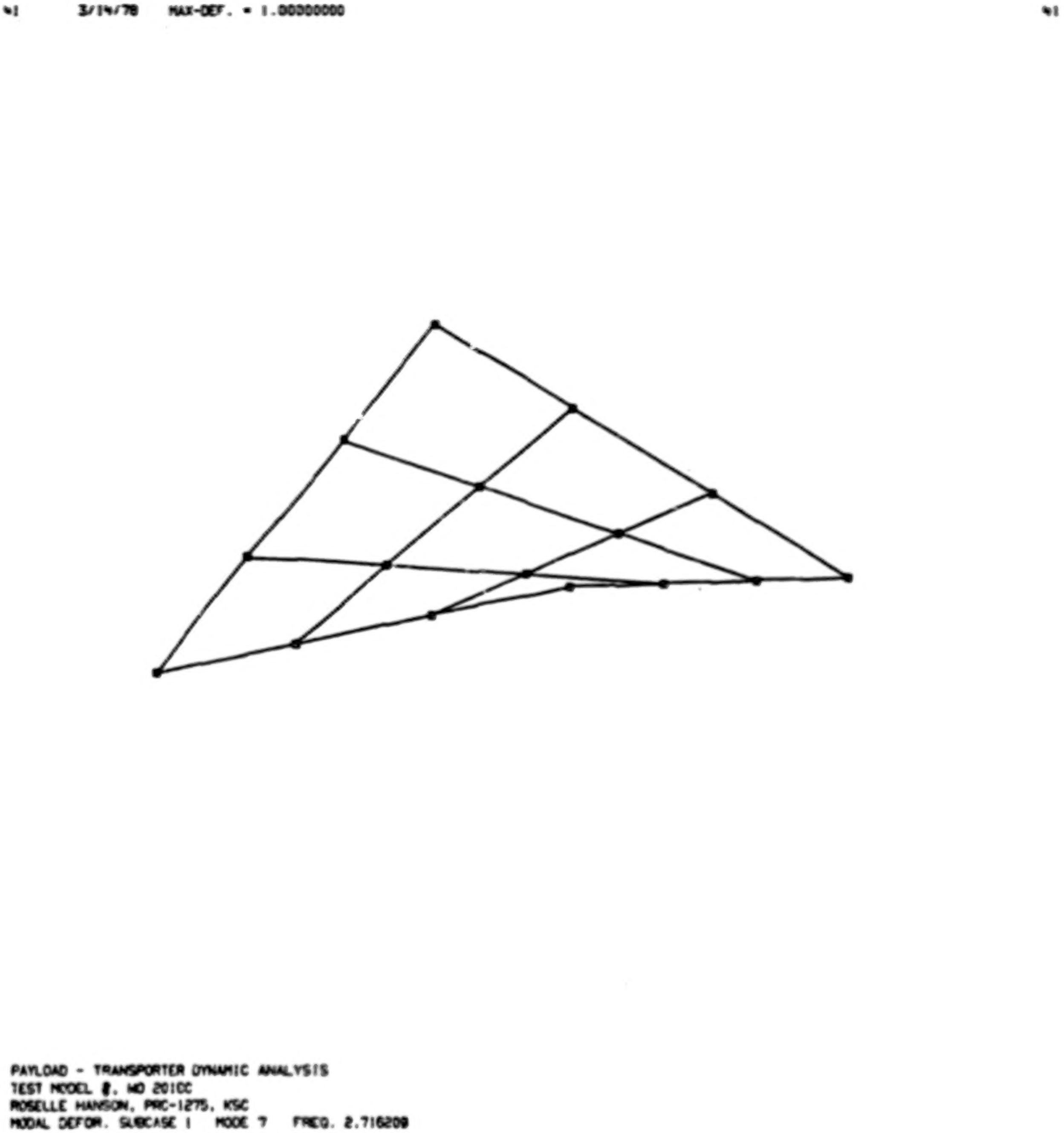
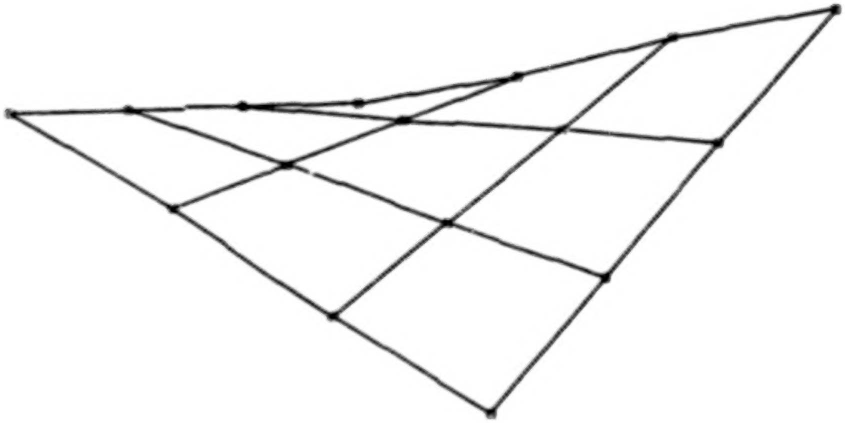

41
3/14/78 MAX-DEF. = 1.00000000
41
PAYLOAD - TRANSPORTER DYNAMIC ANALYSIS
TEST MODEL 1, WO 201CC
ROSELLE HANSON, PRC-1275, KSC
MODAL DEFOR. SUBCASE 1 MODE 7 FREQ. 2.716209

Figure 4. First Torsional Mode, Test Model 1



41 3/17/78 MAX-DEF. = 1.00000000 41
PAYLOAD - TRANSPORTER DYNAMIC ANALYSIS
TEST MODEL 2, WO 2018C
ROSELLE HANSON, PRC-1275, KSC
MODAL DEFOR. SUBCASE 1 MODE 7 FREQ. 13.55737

Figure 5. First Torsional Mode, Test Model 2

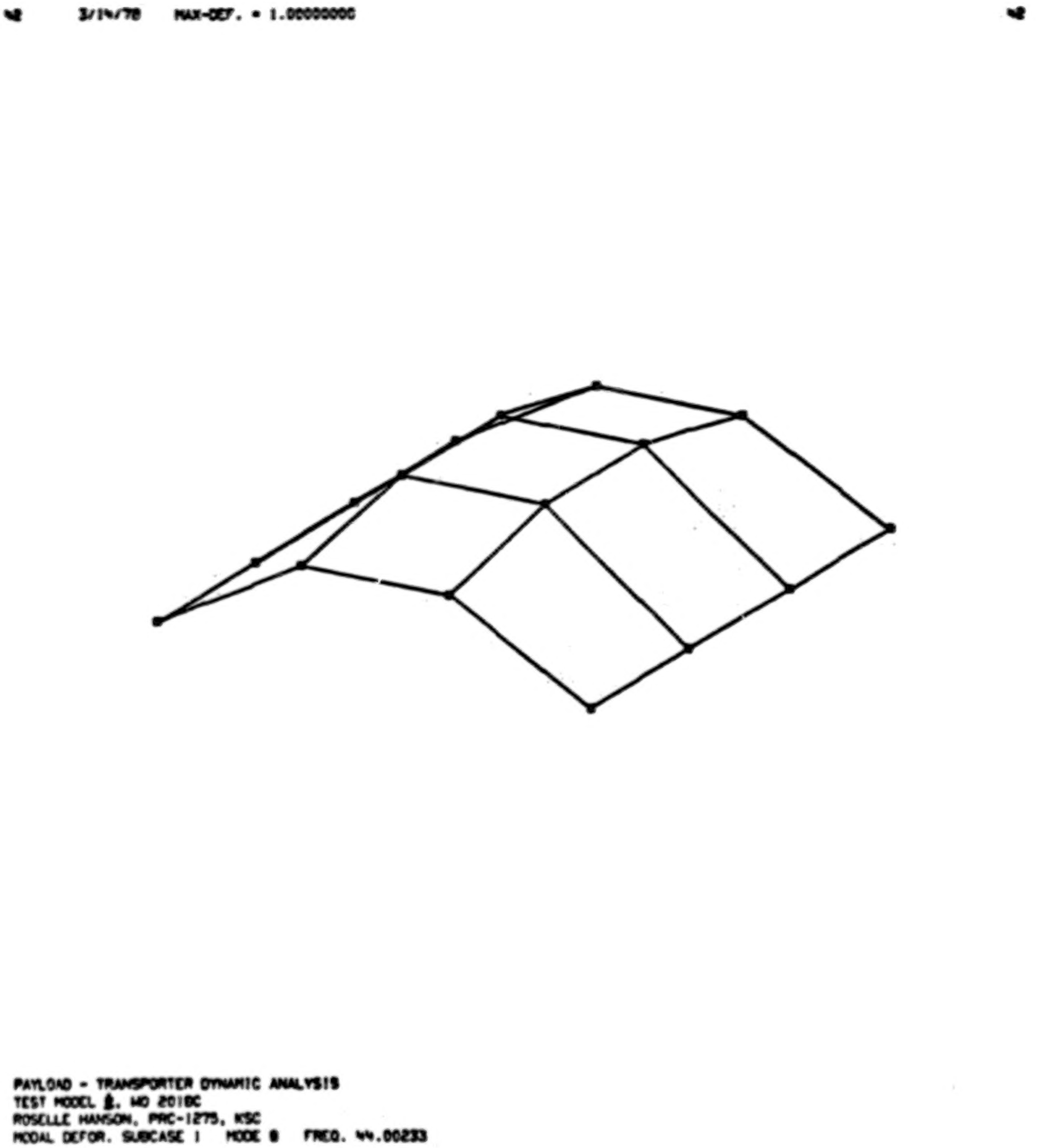
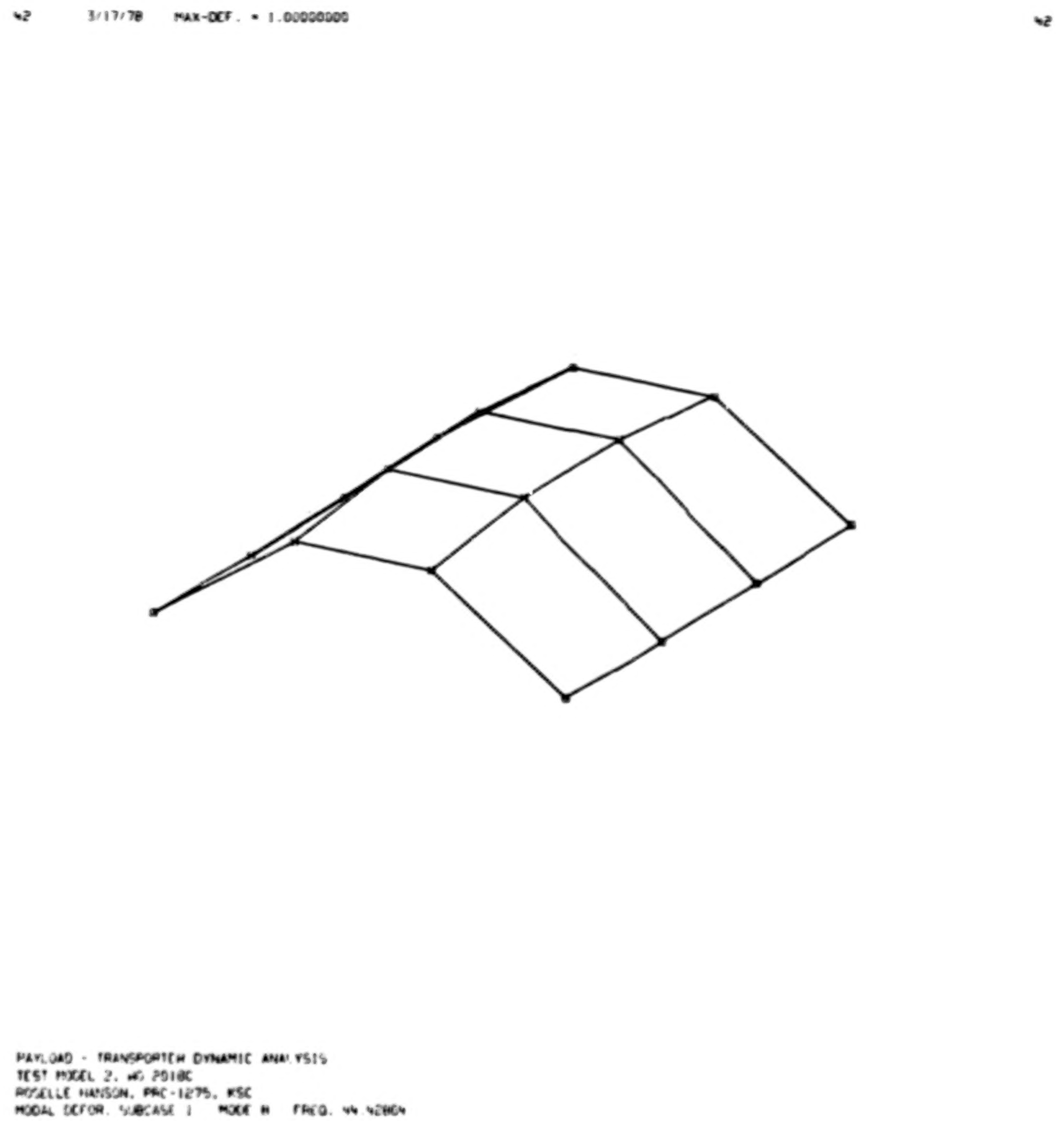

42 3/14/78 MAX-DEF. = 1.00000000 42
PAYLOAD - TRANSPORTER DYNAMIC ANALYSIS
TEST MODEL 8, WO 2018C
ROSELLE HANSON, PRC-1275, KSC
MODAL DEFOR. SUBCASE 1 MODE 8 FREQ. 44.00233

Figure 6. First Bending Mode, Test Model 1



42 3/17/78 MAX-DEF. = 1.00000000
42
PAYLOAD - TRANSPORTER DYNAMIC ANALYSIS
TEST MODEL 2, WO 2018C
ROSELLE HANSON, PRC-1275, KSC
MODAL DEFOR. SUBCASE 1 MODE 8 FREQ. 44.42804

Figure 7. First Bending Mode, Test Model 2

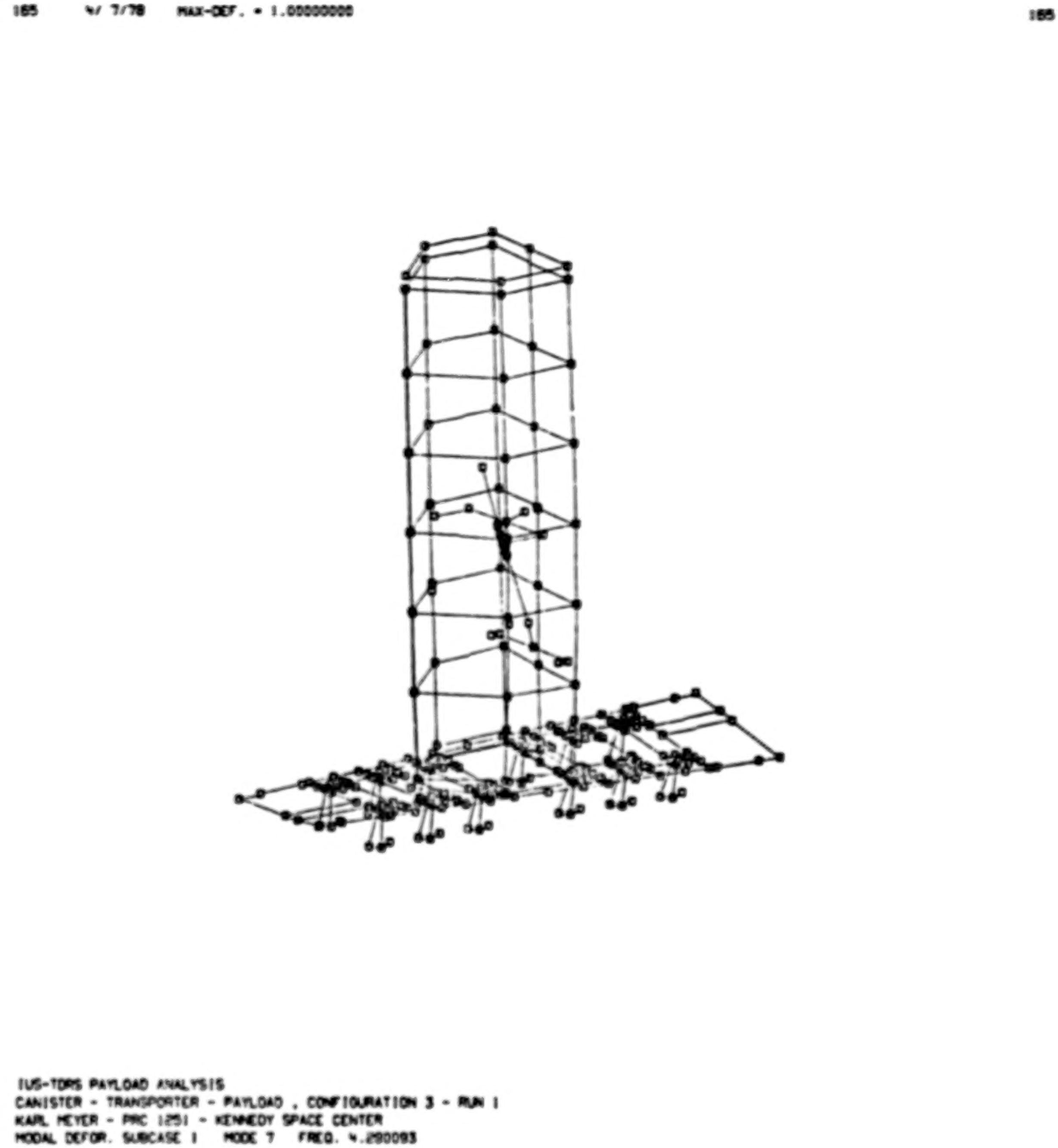

165 4/ 7/78 MAX-DEF. = 1.00000000
165
IUS-TDRS PAYLOAD ANALYSIS
CANISTER - TRANSPORTER - PAYLOAD , CONFIGURATION 3 - RUN 1
KARL MEYER - PRC 1251 - KENNEDY SPACE CENTER
MODAL DEFOR. SUBCASE 1 MODE 7 FREQ. 4.290093

Figure 8. IUS/TDRS Mode

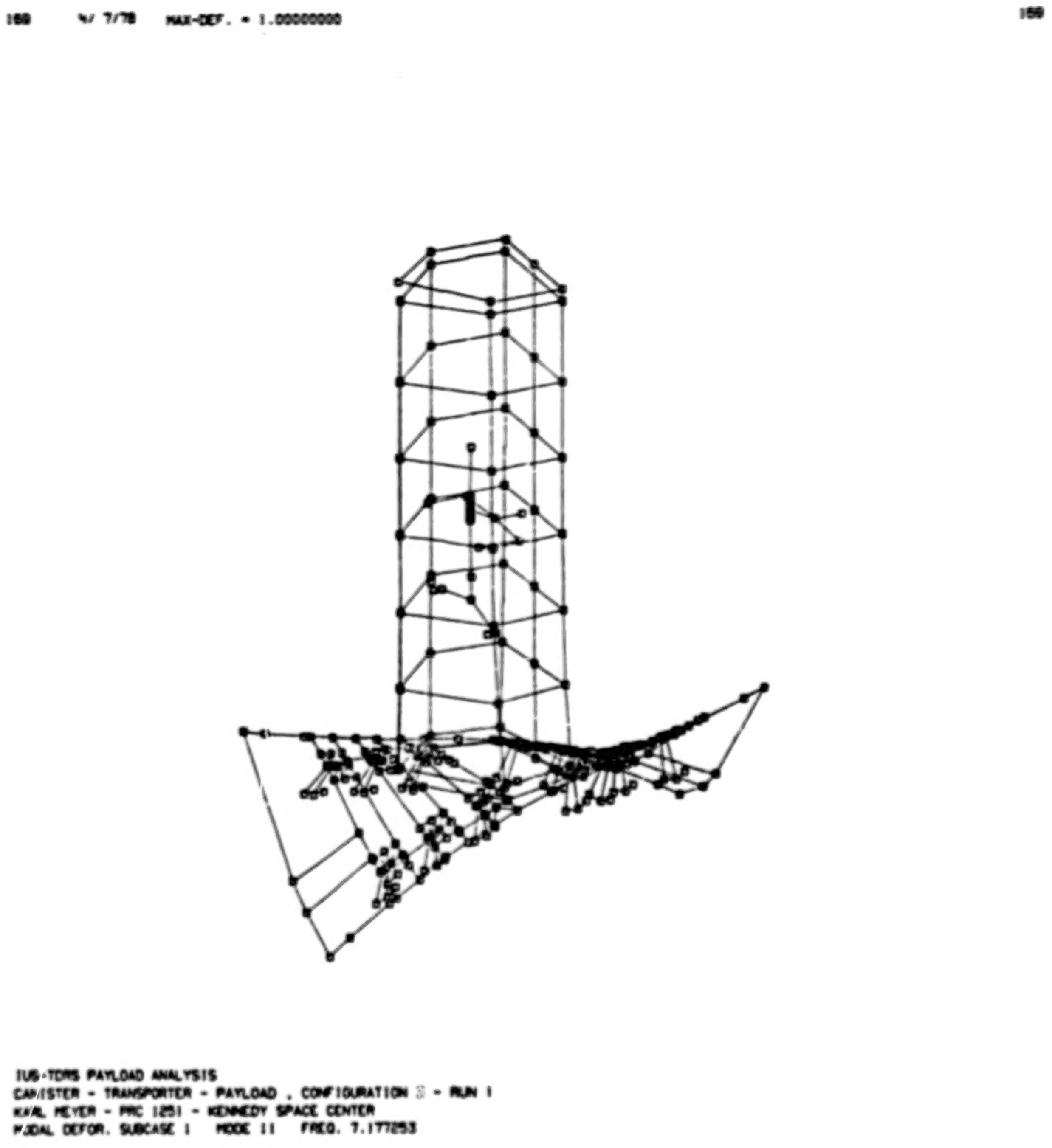

169 4/ 7/78 MAX-DEF. = 1.00000000
169
IUS-TDRS PAYLOAD ANALYSIS
CANISTER - TRANSPORTER - PAYLOAD , CONFIGURATION 3 - RUN 1
KARL MEYER - PRC 1251 - KENNEDY SPACE CENTER
MODAL DEFOR. SUBCASE 1 MODE 11 FREQ. 7.177253

Figure 9. Transporter Torsional Mode

REFERENCES

1. The NASTRAN User's Manual, NASA-SP-222(03), March 1976

2. Torsional Analysis of Rolled Steel Sections, Bethlehem Steel Corporation

BLANK PAGE

STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED PERIODIC STRUCTURES

Anthony J. Kalinowski
Naval Underwater Systems Center

SUMMARY

The paper treats the general problem of solving for the steady state (time domain) dynamic response (i.e., NASTRAN rigid format-8) of a general elastic periodic structure subject to a phase difference loading of the type encountered in traveling wave propagation problems. Two types of structural configurations are considered; in the first type, the structure has a repeating pattern over a span that is long enough to be considered, for all practical purposes, as infinite; in the second type, the structure has structural rotational symmetry in the circumferential direction. Due to the periodic nature of the structure and the traveling wave characteristics of the loading, one need only "cut out" and subsequently model a typical periodic region of the total structure, wherein appropriate periodic boundary conditions (i.e., unknown forces and displacements are forced equal, except for unknown phase angle, for corresponding points on both cuts) are used along the cuts. The paper presents both the theory and a corresponding set of DMAP instructions which permits the NASTRAN user to automatically alter the rigid format-8 sequence to solve the intended class of problems. The new input to a standard version NASTRAN run is a set of alter cards, PARAM cards, and direct input matrix (DMI) partitioning arrays which are needed for the purpose of partitioning and correspondingly restructuring the internal NASTRAN mass, damping and stiffness matrices. Final results are recovered as with any ordinary rigid format-8 solution, except that the results are only printed for the typical periodic segment of the structure. A simple demonstration problem having a known exact solution is used to illustrate the implementation of the procedure.

SYMBOLS

[B]	Damping matrix of nth periodic substructure
$\{\bar{F}\}$	Total applied force vector
i	$\sqrt{-1}$
$[I^u]$	Diagonal unit identity matrix
I	Number of degrees-of-freedom for interior nodes
[K]	Stiffness matrix of nth periodic substructure

L_p	Spatial period of substructure
L	Number of degrees-of-freedom for left cut nodes
[M]	Mass matrix of n^{th} periodic substructure
R	Number of degrees-of-freedom for right cut
t	Time
{U}	Displacement vector
x	Spatial coordinate
ω	Angular driving frequency
θ, ψ	Angle of incident wave
μ	Phase constant (complex form)
μ^*	Phase constant (real part)

INTRODUCTION

A periodic structure consists of a number of identical substructures, coupled together in identical manners to form the whole system, see for example figure 1a. For such systems, under certain loading conditions, it is often possible to treat only one representative substructure in order to obtain the general response for the whole system. For example, if the loading is exactly the same for all substructures, the latest (and even some earlier) versions of NASTRAN can directly solve this class of problem for both static and steadystate cases (i.e., rigid formats 1 and 8). In the case of steady state dynamics problems (rigid format-8) involving traveling wave propagation type inputs, there is a slightly more general loading condition on each periodic substructure, namely that the loading on each substructure is identical except for a known phase constant μ. More specifically, the relation between the applied force vector $\{F\}_n$ in the n^{th} substructure and the one, $\{F\}_{n+1}$ in the n^{th} substructure is given by

$$\{F\}_{n+1} = e^{\mu} \{F\}_n \tag{1}$$

where μ is a known phase constant. For the class of problems addressed in this paper, the phase constant is a purely imaginary constant, i.e.,

$$\mu = 0.0 + i\mu^* \tag{2}$$

and physically refers to the fact that there is no difference in energy loss in processing results from one substructure to the next.

Brillouin (ref. 1) points out that wave motion in periodic systems have been studied for nearly 300 years, wherein physicists and electrical engineers have worked in this field in problem areas relating to optics, crystals, electrical transmission liner, etc. (Elachi, ref. 2, provides a comprehensive list of 287 references in this field). Applications of this theory to engineering structural analysis and solid mechanics type problems is only recent. References 3, 4, and 5 are typical of analytical solutions to this type problem for simple configurations consisting of beams, grillages and plate structures. References 6 and 7 represent a significantly more general approach to the problem wherein their application of the theory of finite elements enables one to solve a much larger class of problems involving rather arbitrary structures than one could treat by purely analytical techniques. References 6 and 7 appear to restrict themselves to the problem of determining the conditions (i.e., values of the steady state response freqency ω) under which propagating or non-propagating free wave motion will occur in the absence of explicit external driving forces.

In the work presented here, periodic structures with explicit external driving forces satisfying Equations (1) and (2) are applied to each substructure as illustrated in Figure 1a.

If the loading and spatial boundary conditions on each substructure are the same, except for the phase difference, μ^*, (i.e., the loading for two typical points in two neighboring substructures separated by the period length L_p, are the same except for a multiplying factor of $e^{i\mu^*}$) it follows that the response in each substructure is also the same except for the phase difference μ^*. A simple example of such a case is a propagating pressure wave passing across an infinitely long ribbed plate as illustrated in figure 2b (the plate is in air and no air-structure interaction effects are considered). The propagating surface loading wave is given by the formula

$$p = p_0 e^{i(kx + \omega t)} \tag{3}$$

thus the phase difference between any two neighboring substructures is $\mu^* = kL_p$, where L_p is the spatial period of the periodic system, p_0 is the input loading pressure amplitude, ω is the steady state driving frequency, x is the horizontal spatial coordinate, and k is the wave number of the loading wave.

Other examples of the phase constant relative to a particular example are shown in figure 2. In figure 2a, we have a known pressure wave loading propagating parallel with the axis of the ribbed cylindrical shell; here, the phase constant μ^* is analogous to the figure 2b example and needs no further explanation. In figure 2c, the incident wave is incident at an oblique angle θ and is incorporated into the formula for the phase constant given in the figure 1c.

There is a special case where the ends of the structure do not extend to infinity (i.e., the ends never meet) but instead are connected cyclically, as in figure 3 for example. For such cyclical cases, μ^* must satisfy an additional constraint, namely $\mu^* = 2\pi/n$ where $n = 1, 2, 3, \cdots$. For example, in the figure 3 case, $n = 8$, thus $\mu^* = \pi/4$.

We limit ourselves to problems having "one-dimensional periodicity", whereby this term we imply that only two cuts are needed (we shall refer to these as the left and right cuts, see figure 1b) to separate the typical substructure from the system. The response within such a substructure can be, however, multi-dimensional. The remainder of the paper focuses on the procedure for obtaining the displacement and stress response within one typical block of the periodic system. The typical substructure block can be made up of various types of structural elements (including both elements with structural damping and nodes with scalar dampers attached) contained within the NASTRAN library of elements (e.g., CQDMEM, CQDMEM1, CBAR, CONROD ... etc.).

SOLUTION FORMULATION

The solution procedure presented here is very similar to the one in reference (6), except for the fact that here we are considering problems with explicit forcing functions. The first step in the solution procedure is to "cut out" the typical substructure from the overall periodic structure as illustrated in figure 1b and to subsequently replace the cut nodes with the internal forces ($\{\overline{F}^c_\ell\}_n$, for the left cut and $\{\overline{F}^c_r\}_n$ for the right cut) that existed at those nodes before cutting. The displacements at the cut nodes are similarly denoted by $\{\overline{U}_\ell\}_n$ and $\{\overline{U}_r\}_n$ where subscripts ℓ and r denote left and right and the subscript n denotes the n^{th} substructure. Since we are only focusing on the results for the n^{th} substructure, it is convenient to drop the subscript n from here on for notational convenience.

The governing equations of motion for the substructure are first expressed in the familiar finite element form

$$[M]\{\ddot{\overline{U}}\} + [B]\{\dot{\overline{U}}\} + [K]\{\overline{U}\} + \{\overline{F}\} \qquad (4)$$

where $[M]$, $[B]$, $[K]$ are the mass, damping and stiffness matrices of the n^{th} substructure respectively; $\{\overline{U}\}$ is the displacement vector of all nodes of the n^{th} substructure; $\{\overline{F}\}$ is the generalized force fector; $(\dot{\ }) \equiv d(\ )/dt$, and bars above the variables denote the fact that the variables are complex and that the harmonic time response $e^{i\omega t}$ has not yet been suppressed thus

$$\{\overline{U}\} = \{U\}e^{i\omega t} \qquad \{\overline{F}\} = \{F\}e^{i\omega t} \qquad (5)$$

The next major step is to partition the matrices and vectors of equation (4) into left cut unknowns, right cut unknowns and interior unknowns (subscripts ℓ, r, i refer to left, right and interior respectively and L, R, I refer to the total number of displacement component unknowns for the left, right and interior domain respectively; note due to periodicity, L = R. Thus it follows that after partitioning we have

$$[M] = \underset{(L+R+I) \times (L+R+I)}{\left[\begin{array}{c|c|c} M_{\ell\ell} & M_{\ell i} & M_{\ell r} \\ \hline M_{i\ell} & M_{ii} & M_{ir} \\ \hline M_{r\ell} & M_{ri} & M_{rr} \end{array}\right]} \qquad [B] = \left[\begin{array}{c|c|c} B_{\ell\ell} & B_{\ell i} & B_{\ell r} \\ \hline B_{i\ell} & B_{ii} & B_{ir} \\ \hline B_{r\ell} & B_{ri} & B_{rr} \end{array}\right]$$

$$[K] = \left[\begin{array}{c|c|c} K_{\ell\ell} & K_{\ell i} & K_{\ell r} \\ \hline K_{i\ell} & K_{ii} & K_{ir} \\ \hline K_{r\ell} & K_{ri} & K_{rr} \end{array}\right] \qquad \{\overline{U}\} = \underset{(1) \times (L+R+I)}{\left\{\begin{array}{c} \overline{U}_{\ell} \\ \hline \overline{U}_{i} \\ \hline \overline{U}_{r} \end{array}\right\}} \tag{6}$$

$$\{\overline{F}\} = \left\{\begin{array}{c} \overline{F}^{c}_{\ell} \\ \hline 0 \\ \hline \overline{F}^{c}_{r} \end{array}\right\} + \left\{\begin{array}{c} \overline{F}^{a}_{\ell} \\ \hline \overline{F}^{a}_{i} \\ \hline \overline{F}^{a}_{r} \end{array}\right\}$$

Note the generalized force vectors $\{\overline{F}\}$ has been further decomposed as the sum of an unknown force vector, $\{\overline{F}^{c}\}$, (which denotes the yet unknown internal forces existing at the cuts in the structure) and a known applied force vector, $\{\overline{F}^{a}\}$, (which denotes all known forces existing within and at the cuts of the periodic substructure).

The full periodic structure is cut (left and right cuts), therefore it follows that the internal nodal forces normally existing at the cuts now play the role of external (as yet unknown) applied forces.

The special case of an externally applied force appearing at a left or right cut requires special attention in that one component of the total force vector is due to the externally applied force and the other component is due to the internal force at the cut. The external force value on a cut must be shared between the generic substructure block being analyzed and its immediate neighbor; consequently these end type external force values are divided in half (see, for example, the situation in figure 1c where node 1 lies on the left cut).

A further relation that is needed in the formulation relates to the fact that the right end of the n^{th} generic substructure is the beginning (left end) of the n+1 generic substructure, thus from Equation (1) it follows that

$$\{\bar{F}_r^c\} = -e^{\mu}\{\bar{F}_\ell^c\}$$

$$\{\bar{U}_r\} = e^{\mu}\{\bar{U}_\ell\} \qquad (7)$$

where the minus sign in the first of Equation (7) accounts for the fact that internal nodal forces acting as external forces on the right cut of generic substructure n are opposite in sign to the internal nodal forces acting on the left cut neighboring substructure n + 1.

The next step in the development is to substitute Equations (5) and (6) into Equation (4); the subsequent cancellation of $e^{i\omega t}$ permits us to drop the bar superscript notation thus arriving at a "reduced form of Equation (4)". At this point there are five groups of unknowns, namely $\{U_r\}$, $\{U_\ell\}$, $\{U_i\}$, $\{F_r^c\}$, $\{F_\ell^c\}$. The three row partitions of the reduced Equation (4) in conjunction with the two Equations (7), provide 3 + 2 = 5 corresponding groups of equations to balance the five groups of unknowns. Next, we substitute Equations (7) into the reduced form of Equation (4), and subsequently employ the third row partition of reduced Equation (4) to eliminate the $\{F_\ell^c\}$ unknown. Doing these operations result in the following set simultaneous equations for the displacement unknowns $\{U_\ell\}$ and $\{U_i\}$:

CONTENTS

Page

FOREWORD . . . . iii 1/A5

1. ON THE CONGRUENT FEATURE IN NASTRAN 1 1/A9
by P. R. Pamidi (Computer Sciences Corporation)

2. DEVELOPMENT OF THE LEARJET 28/29 WING FOR NASTRAN ANALYSIS. 11 1/B6
by Robert J. Boroughs (Gates Learjet Corporation)

3. TENSIONING OF A BELT AROUND A DRUM USING MEMBRANE ELEMENTS 33 1/E1
by C. H. S. Chen (B. F. Goodrich Company)

4. THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS SUBJECTED TO ASYMMETRIC THERMAL LOADS 41 1/E12
by Gordon R. Negaard (ANAMET Laboratories, Inc.)

5. STRESS CONCENTRATIONS IN SCREW THREADS. 65 2/B1
by G. Peter O'Hara (U. S. Army Armament Research and Development Command)

6. CONDENSING LOADED POINTS FOR TRANSIENT IN SUBSTRUCTURING. 79 2/C6
by Thomas G. Butler (Butler Analyses)

7. IMPLEMENTATIONS OF A TRAPEZOIDAL RING ELEMENT IN NASTRAN FOR ELASTIC-PLASTIC ANALYSIS. 101 2/E5
by P. C. T. Chen and G. P. O'Hara (U. S. Army Armament Research and Development Command)

8. NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT 113 2/F5
by Karl A. Meyer (Planning Research Corporation)

9. STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED PERIODIC STRUCTURES 131 2/G9
by Anthony J. Kalinowski (Naval Underwater Systems Center)

10. APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS AT NORTHROP 165 3/C4
by Ashok K. Singh (Northrop Corporation)

11. IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS OPERATING SYSTEM. 189 3/F2
by Stephen S. Britten and Betsy Schumacker (M.I.T.)

MICROFILMED FROM BEST
AVAILABLE COPY

Page

12. IFEMS - AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM 205
by Spencer McKellip, Todd Schuman, and Spencer Lauer (Sikorsky Aircraft) 3/G5

13. APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL. 223
by James L. Rogers, Jr. (NASA Langley Research Center) 4/B6

$$\underbrace{\left[\begin{array}{c|c} -\omega^2\ [MLL] + & -\omega^2\ [MLI] + \\ i\omega\ [BLL] + & i\omega\ [BLI] + \\ [KLL] & [KLI] \\ \hline -\omega^2\ [MIL] + & -\omega^2\ [MII] + \\ i\omega\ [BIL] + & i\omega\ [BII] + \\ [KIL] & [KII] \end{array}\right]}_{(L+I)\ \text{x}\ (L+I)\ \text{matrix}} \cdot \left\{\begin{array}{c} \{U\}_\ell \\ --- \\ \{U_i\} \end{array}\right\} = \underbrace{\left\{\begin{array}{c} \{F_\ell^a\} + e^{-\mu^* i}\{F_r^a\} \\ -------- \\ \{F_i^a\} \end{array}\right\}}_{(1)\ \text{x}\ (L+I)\ \text{vector}} \tag{8}$$

where

$$\begin{aligned}
[MLL] &= [M_{\ell\ell}] + \text{Cos}\mu^*\ [M_{\ell r}] + \text{Cos}\mu^*\ [M_{r\ell}] + [M_{rr}] \\
[BLL] &= [B_{\ell\ell}] + \text{Cos}\mu^*\ [B_{\ell r}] - \omega\text{Sin}\mu^*\ [M_{\ell r}] + (\text{Sin}\mu^*/\omega)\ [K_{\ell r}] \\
&\quad + \text{Cos}\mu^*\ [B_{r\ell}] + [B_{rr}] + \omega\text{Sin}\mu^*\ [M_{r\ell}] - (\text{Sin}\mu^*/\omega)\ [K_{r\ell}] \\
[KLL] &= [K_{\ell\ell}] + \text{Cos}\mu^*\ [K_{\ell r}] - \omega\text{Sin}\mu^*\ [B_{\ell r}] \\
&\quad + \text{Cos}\mu^*\ [K_{r\ell}] + [K_{rr}] + \omega\text{Sin}\mu^*\ [B_{r\ell}] \\
[MLI] &= [M_{\ell i}] + \text{Cos}\mu^*\ [M_{ri}] \\
[BLI] &= [B_{\ell i}] + \text{Cos}\mu^*\ [B_{ri}] + \omega\text{Sin}\mu^*\ [M_{ri}] - (\text{Sin}\mu^*/\omega)\ [K_{ri}] \\
[KLI] &= [K_{\ell i}] + \text{Cos}\mu^*\ [K_{ri}] + \omega\text{Sin}\mu^*\ [B_{ri}] \\
[MIL] &= [M_{i\ell}] + \text{Cos}\mu^*\ [M_{ir}] \\
[BIL] &= [B_{i\ell}] + \text{Cos}\mu^*\ [B_{ir}] - \omega\text{Sin}\mu^*\ [M_{ir}] + (\text{Sin}\mu^*/\omega)\ [K_{ir}] \\
[KIL] &= [K_{i\ell}] + \text{Cos}\mu^*\ [K_{ir}] - \omega\text{Sin}\mu^*\ [B_{ir}] \\
[MII] &= [M_{ii}] \\
[BII] &= [B_{ii}] \\
[KII] &= [K_{ii}]
\end{aligned} \tag{9}$$

At this point, the linear set of complex algebraic Equations (8) can be solved for the unknown displacements $\{U_\ell\}$, $\{U_i\}$. The unknown displacement at the right cut, $\{U_r\}$ can be easily computed with the second of equation (7). The size of the algebraic system is governed by the (L+I) x (L+I) coefficient matrix (i.e., matrix multiplying the unknown displacement vector) where L+I equals the number of left cut, L, plus interior, I, unknown displacement components. Typically I>>L, therefore it is not a very big additional burden on the equation solver to include the second of Equation (7) as part of the overall system (actually, we add R extra unknowns, $\{U_r\}$, and R extra equations (where R = number of right cut unknowns). Thus in place of Equation (8), we consider the slightly larger, but equivalent system of

$$\left[\begin{array}{c|c|c} -\omega^2\ [MLL]\ + & -\omega^2\ [MLI]\ + & \\ i\omega\ [BLL]\ + & i\omega\ [BLI]\ + & [0] \\ {[KLL]} & {[KLI]} & \\ \hline -\omega^2\ [MIL]\ + & -\omega^2\ [MII]\ + & \\ i\omega\ [BIL]\ + & i\omega\ [BII]\ + & [0] \\ {[KIL]} & {[KII]} & \\ \hline i\omega\ [BRL]\ + & & \\ & [0] & [KRR] \\ {[KRL]} & & \end{array}\right] \cdot \left\{\begin{array}{c} \{U_\ell\} \\ \hline \{U_i\} \\ \hline \{U_r\} \end{array}\right\} = \left\{\begin{array}{c} \{F^a_\ell\}\ + \\ e^{-\mu^* i}\{F^a_r\} \\ \hline \{F^a_i\} \\ \hline \{0\} \end{array}\right\} \qquad (10)$$

(L+I+R) x (L+I+R)
coefficient matrix

where
$$\begin{aligned} {[BRL]} &= (\sin\mu^*/\omega)\ [I^u] \\ {[KRL]} &= \cos\mu^*\ [I^u] \\ {[KRR]} &= -[I^u] \end{aligned} \qquad (11)$$

and $[I^u] \equiv$ diagonal unit identity matrix

The RI block of the displacement coefficient matrix in Equation (10) above is identically zero, thus the bottom R rows of the system of simultaneous equations are totally independent of the solution to the top L+I rows. The length of the solution vector L+I+R is of exactly the same length of the original substructure matrix (Equations (4) and (6)), consequently the modification of the DMAP instructions becomes simplier because of the fact that one need only intercept the logic of the equation solver and replace the existing mass stiffness and damping matrices with the modified matrices defined by the new coeffient matrix of Equation (10) (and associated new entry definitions from Equations (9) and (11)). Since the length of the solution vector is still the same as the original problem before modification, the post processing DMAP operations for displacement printout, stress recovery, etc. need not be modified.

An alternate scheme (although not yet implemented) would be to modify the input to the complex equation solver to accept the smaller (L+I) x (L+I) coeffient matrix used in Equation (8) directly. After solving the smaller L+I length displacement vector, the full vector (i.e., attaching the missing $\{U_r\}$ portion) can be formed by expanding it to length L+I+R via the second of Equation (7). Finally, stress and displacement results can be processed in the usual way with existing DMAP operations.

RIGID FORMAT-8 DMAP MODIFICATION FOR NASTRAN

The periodic structure capability described in the previous section can be implemented in a standard version of NASTRAN. In particular, the DMAP sequence required to perform the necessary operations are listed in Appendix A. This DMAP sequence was checked out on an 1108 computer, standard version of level 15.5 NASTRAN and is introduced in the EXECUTIVE CONTROL deck with the following instructions:

```
ALTER 138

(see Appendix A for specific instructions)
ALTER 139,139

(replace KDD, BDD, MDD with KDDX, BDDX, MDDX
  within call arguments of FFRD module              level 15.5
  see Appendix A for detailed instruction)          implementation

ALTER 140

(Conditional print statement, see Appendix-
  A for detailed DMAP instructions)
```

These same level 15.5 DMAP instructions can also be applied to level 17.0 NASTRAN, the only difference being that

replace ALTER 138 with ALTER 158

replace ALTER 139,139 with ALTER 159,159

replace ALTER 140 with ALTER 160

It is pointed out, however, that the level 17.0 modifications described above have not actually been tried although due to the similarity of the change, the DMAP sequence is expected to work.

It is important to note that we are modifying the standard NASTRAN unknown displacement vector coefficient matrix just prior to the entry into the FFRD module used for the solution to the simultaneous complex algebraic equations. The implication of this statement is that the row numbering scheme for the displacement vector has already accounted for the fact that single point constraints, multipoint constraints and omitted coordinates have already been accounted for. Thus, for example, the length of the $\{U_\ell\}$ vector, L, is not simply the number of nodes on the left cut times the degrees-of-freedom per node, but rather is less by the amount corresponding to the number of SPC's, MPC's and OMIT's relating to the nodes along the left cut. Similar comments apply to the length of the $\{U_i\}$ and $\{U_r\}$ vectors. The understanding of the above displacement vector length comments must be clearly understood by the user before attempting to fill out the input data matrix partitioning vectors CV100, CV010, CV001 defined later in this paper.

INPUT DATA FOR NASTRAN RUN

The BULK DATA input to a typical periodic structure run consist of two basic parts. The first part corresponds to the usual bulk data input cards normally required to make a NASTRAN run (e.g., GRID CARDS, ELEMENT CARDS, DAREA CARDS, FREQ CARD, DLOAD CARD, etc.); the second part consists of special input cards that are explained in the following text.

PARAM Cards

These cards are used to enter various matrix coefficients appearing in Equations (9) and (11); especially the constants

Text Variable	Computer Variable
$\mathrm{Cos}\mu^*$	≡ CMST
$0.0 + i\omega$	≡ FIØMEG
$-\omega^2$	≡ NØMEG2
-1.0	≡ NØNE
$-\sin\mu^*/\omega$	≡ NSMSBØ
$-(\sin\mu^*)\omega$	≡ NSMSTØ
$+1.0$	≡ PØNE
$\sin\mu^*/\omega$	≡ SMSBØ
$(\sin\mu^*)\omega$	≡ SMSTØ

(12)

note: 0 = zero
Ø = letter

are read in on standard NASTRAN PARAM cards where μ^* is the phase angle defined in Equation (2) and ω is the angular driving frequency in radians per second ($\omega = f \cdot 2\pi$ where f = driving frequency in H_z specified on the FREQ card). The format for a typical PARAM card is:

Col's 1 - 8	PARAM
Col's 9 - 16	one of the 8 computer variable names defined by Equation (12)
Col's 17 - 24	real part of variable defined in Col's 9-16
Col's 25 - 32	imaginary part of variable defined in Col's 9-16 (only non zero entry is for variable FIØMEG)

Comments

Strictly speaking, the real part of variable FIØMEG should be 0.0; however, for the NUSC Univac 1108, operating with the level 15.5 version of NASTRAN used to implement the procedure, an arbitrary small number if entered (say $1.0\text{x}10^{-20}$) in order to avoid a strange system type error message that is printed when exactly 0.0 is entered as the real part. The FIØMEG variable is only used to compute and subsequently print out the internal forces at the cuts after all the main calculations for displacement are completed. The mentioned error message probably will not appear if other NASTRAN versions and/or other computer systems are used.

DMI Cards for Matrix Partitioning

A set of DMI direct matrix input cards are needed to provide the information NASTRAN needs to partition the mass, damping and stiffness matrices. Three groups of cards are needed; a column partitioning vector for the left cut group of displacement node components, CV100; a column partitioning vector for the interior group of displacement node components, CV010; and a column partitioning vector for the right cut group of displacement node components, CV001. A set of row partitioning vectors are automatically generated by the Appendix A DMAP instructions. The column partitioning are made up of entries that are either 1.0 or 0.0. Since all entries within NASTRAN are assumed to be zero unless otherwise specified, the user need only enter 1.0 values in the appropriate slot in each of the above mentioned partitioning vectors. The rules are simple and are as follows:

- Formation of left cut partitioning vector CV100

  Enter a 1.0 in each row number corresponding to each active independent component degrees-of-freedom lying along the left cut. The length of the CV100 vector is L+I+R and there should be L 1.0 entries (the remaining I+R entries are automatically zero by virtue of not being defined). If the left cut nodal numbering pattern is sequential and starts with the lowest node number of the whole system (e.g., node 1, 2, 3, . . .), then the first L entries of CV100 will be all 1.0 values. However, if the left cut numbering scheme does not contain only the lowest node numbers, but instead the whole system is numbered at random, then the L 1.0 entries will correspondingly be distributed throughout the CV100 vector, and the "bookkeeping" involved with defining the CV100 vector becomes messy. The user having MPC's, SPC's or OMIT's applied to nodes along the left cut must be sure to account for these during the process of entering the 1.0 values into the partitioning vector CV100.

- Formation of the interior partitioning vector CV010

  Enter a 1.0 in each row number corresponding to each active independent component degree-of-freedom lying on the interior of the structure. The length of the CV010 vector is L+I+R and there should be I 1.0 entries (the remaining L+R entries are automatically zero). If the interior nodes are numbered sequentially, (following the same sequential pattern used in the CV100 vector), then the middle L+1, L+2, . . . L+I entries of the CV010 vector will all be 1.0 values. Again remember to account for SPC's, MPC's and OMIT's in the numbering scheme.

- Formation of the right cut partitioning vector CV001

  Enter a 1.0 in each row number corresponding to each active independent component degree-of-freedom lying on the right cut of the periodic structure. The length of the CV001 vector is L+I+R and there should be R 1.0 entries (the remaining L+I entries are automatically zero). If the left cut, interior, and right cut nodes are all numbered sequentially (in the respective order mentioned), then the ending L+I+1, L+I+2, . . . L+I+R entries of the CV001 vector will all be 1.0 values. Again remember to account for SPC's, MPC's and OMITS's in the numbering scheme.

DMI Cards Format

The bulk data cards for the definitions of the partitioning vectors via the standard DMI cards is as follows:

- CV100 vector cards

Col's	Entry
	first header card
1-8	DMI
9-16	CV001
17-24	0

Col's	Entry
25-32	2
33-40	1
41-48	1
49-56	blank
57-64	integer value equal to magnitude of (L+R+I)
65-72	1
	second card
1-8	DMI
9-16	CV001
17-24	1
25-32	enter integer row number ,say N1, of first 1.0 entry
33-40	1.0 for entry N1
41-48	1.0 or 0.0 for entry N1+1
49-56	1.0 or 0.0 for entry N1+2
57-64	1.0 or 0.0 for entry N1+3
65-72	1.0 or 0.0 for entry N1+4
73-80	Continuation card name, say CONT1, if needed.
	third continuation card (if needed)
1-8	+CNT1
9-16	1.0 or 0.0 for entry N1+ 5
17-24	• •
25-32	• •
33-40	• •
41-48	• •
49-56	• •
57-64	• •
65-72	1.0 or 0.0 for entry N1+12
73-80	Continuation card name, say CONT2, if needed.
	fourth continuation card (if needed)
	similar to third continuation card ... etc.

• CV010 Vector Cards

These cards are made up analogously to the CV100 vector already described above, the only differences being that on the first two cards, replace CV100 with CV010; also, the number of the first 1.0 entry slot (N1 value in col's 25-32 of the second CV100 data card) is different.

• CV001 Vector Cards

These cards are made up analogously to the CV100 vector already described above, the only differences being that on the first two cards, replace CV100 with CV001; also the number of the first 1.0 entry slot (N1 value in col's 25-32 of the second CV100 data card) is different.

Finally, examples of making the partitioning cards is given later in a demonstration problem.

DMI Cards for Merge Operations

A set of DMI direct matrix input cards are needed to define certain dummy matrices which NASTRAN needs to successfully merge certain internal matrices within the Appendix-A DMAP sequence. These input cards will consist of a group of eight cards that must always be included for a run. The only thing that changes from one NASTRAN run to another is the lengths of these dummy arrays. These dummy null arrays are only introduced to avoid DMAP error message printouts for Univac 1108, level 15.5 NASTRAN that occurred when the lead matrix entry in the standard DMAP MERGE operation is not defined.

DMI Cards Format

Col's	Entry
	first header card
1-8	DMI
9-16	LIXLI enter characters, (does not mean multiply LI times LI)
17-24	0
25-32	2
33-40	1
41-48	1
49-56	blank
57-64	enter integer equal to L plus I
65-72	enter integer equal to L plus I
	second card
1-8	DMI
9-16	LIXLI
17-24	1
25-32	1
33-40	0.0
	third card
	same as first card, except
Col's 9-16	enter LIXL2
Col's 57-64	enter integer equal to L plus I
Col's 65-72	enter integer equal to L plus L
	fourth card
	same as second card, except
Col's 9-16	enter LIXL2

	fifth card
	same as first card, except
Col's 9-16	enter L2XLI
Col's 57-64	enter integer equal to L plus L
Col's 65-72	enter integer equal to L plus I
	sixth card
	same as second card, except
Col's 9-16	enter L2XLI
	seventh card
	same as first card, except
Col's 9-16	enter L2XL2
Col's 57-64	enter integer equal to L plus L
Col's 65-72	enter integer equal to L plus L
	eighth card
	same as second card, except
Col's 9-16	enter L2XL2

DMI Cards for Unit Matrix Definition

Finally, a set of DMI direct matrix input cards as needed to define the unit matrix $[I^u]$ employed in Equation (11).

DMI Cards Format

Col's	Entry
	first header card
1-8	DMI
9-16	UMATR
17-24	0
25-32	6
33-40	1
41-48	2
49-56	blank
57-64	enter integer equal to L plus L
64-72	enter integer equal to L plus L

Col's	Entry
	second card
1-8	DMI
9-16	UMATR
17-24	1
25-32	1
33-40	1.0
	third card
	same as second card, except
Col's 17-24	enter 2
Col's 25-32	enter 2
	fourth card
	same as second card, except
Col's 17-24	enter 3
Col's 25-32	enter 3
	.
	.
	etc.
	.
	.
	last ($L+1^{st}$) card
	same as second card, except
Col's 17-24	enter integer value equal to L
Col's 25-32	enter integer value equal to L

DEMONSTRATION PROBLEM

The use of the DMAP sequence in conjunction with the new PARAM and DMI input cards defined in the previous section will perhaps be better understood by including a specific example solution that has the features that (1) is a small size problem convenient for matrix operation checking and debugging purposes; (2) contains most of the main ingredients typical of a representative problem, thus the system has mass, stiffness, and damping; (3) the exact solution to the problem is known for checking purposes. The problem illustrated in figure 4a meets all of these conditions, and corresponds to a plane pressure wave propagating in an infinite acoustic fluid medium. Upon sampling the pressure response along any two parallel vertical cuts (separated by the horizontal distance L_p) it can be shown that the response (pressure and displacement) is different only by the amount $e^{i\mu^*}$ where for this particular problem

$$\mu^* = \left(\frac{\omega}{c}\right)(L_p)\ \mathrm{Cos}\psi \qquad (13)$$

where ω = driving frequency (rad/sec), L_p = length (in.) between two parallel cuts, c = compressional wave speed (in/sec); ψ = angle of incident wave. The solution for the pressure and motion response is sought for the shaded region shown in figure 4a. Only the dotted outlined region is modeled with finite elements and is correspondingly shown in figure 4b.

In order to exercise this demonstration problem to the fullest extent, we will use different types of boundary conditions on all four sides of the figure 4b finite element model.

Boundary Conditions

- Left and right vertical cuts

  Since this problem falls within the class of problems solvable by the phase difference type boundary condition, boundary conditions specified by Equations (7) are enforced. Here the left and right cut forces and displacements are taken as unknowns.

- Top face

  The boundary condition for the top face is different from the left and right face in that here we explicitly apply the free field pressure (converted into equivalent nodal forces). The formula for the freely propagating pressure wave is given by the expression

$$\overline{p}(x,y,t) = \underbrace{p_0 e^{i(kr)}}_{p(x,y)} e^{i\omega t} \qquad (14)$$

where $k = \omega/c$ = wave number (in.$^{-1}$)

r = spatial coordinate normal to direction of wave propagation (in.)

p_0 = steady state pressure amplitude (psi)

$i = \sqrt{-1}$

p = pressure (psi) (spatial variation)

and r is related to the x,y coordinates by the relation

$$r = x \cos\psi + y \sin\psi \qquad (15)$$

The y-direction force at upper face node 3 is computed by substituting p(x,H) from Equation (14) into the expression

$$({}_3F_\ell^a)_y = \int_{x=0}^{x=L_p/4} - p(x,H)\, dx \qquad (16)$$

the force at upper face node 4 is computed similarly by

$$({}_4F_i^a)_y = \int_{x=L_p/4}^{x=3L_p/4} -p(x,H)dx \tag{17}$$

and finally the upper face node 9 is computed similarly by

$$({}_9F_r^a)_y = \int_{x=3L_p/4}^{x=L_p} -p(x,H)dx \tag{18}$$

The demonstration problem is evaluated for the following specific input data:

$$\begin{aligned}
p_0 &= 100.\ \text{psi}\\
c &= 60000.\ \text{in/sec}\\
L_p &= 2.0\ \text{in.}\\
H &= 2.0\ \text{in.}\\
f &= \omega/2\pi = 3000.\ \text{Hz}\\
\psi &= 45^0\\
\rho &= \text{mass density} = .000096\ \text{lb.-sec/in}^4
\end{aligned} \tag{19}$$

Substituting the above Equation (19) input constant into Equations (16), (17) and (18) results in

$$\begin{aligned}
({}_3F_\ell^a)_y &= -49.8972\ e^{i\ 28.6378^0}\\
({}_4F_i^a)_y &= -99.7944\ e^{i\ 38.1837^0}\\
({}_9F_r^a)_y &= -49.9742\ e^{i\ 47.7297^0}
\end{aligned} \tag{20}$$

An important point must be made regarding loading the final force array $\{F\}$ (Equation (4)) for the periodic structure problem. Observation of the Equation (10) loading vector of the new periodic structure problem statement, reveals that the left cut applied forces, $\{F_\ell^a\}$, are applied, as normal, to the corresponding left cut node; also, the interior applied forces, $\{F_i^a\}$, are applied, as normal, to the corresponding interior nodes; however, the right cut applied forces, $\{F^a_r\}$, are not applied to the right cut nodes, but rather, are first multiplied by the complex constant $e^{-\mu^* i}$, and then applied to the corresponding left cut node. For the demonstration problem at hand, substituting Equation (19) into Equation (13) implies that $\mu^* = 25.45584^o$, thus

$$e^{i\mu^*} \left({}_9F^a_r\right)_y = -49.9742\, e^{i22.27387^o} \tag{21}$$

therefore, in summary, at node number 3 in the y direction apply a net sum

$$\text{force} = -49.8972\, e^{i28.6378^o} - 49.9742\, e^{i22.27387^o} \tag{22}$$

and at node number 4 in the y direction apply a net

$$\text{force} = -99.7944\, e^{i38.1837^o} \tag{23}$$

and at node number 9 in the y direction apply a net

force = 0.0 (i.e., do not apply any force).

- Bottom face

  The boundary condition for the bottom face could have been selected similar to the top face (i.e., we convert the pressure into equivalent nodal forces). However, instead we use a slightly more complicated boundary condition that permits us to introduce a [B] matrix entry into the problem. More specifically, the pressure and normal velocity along the bottom cut can be obtained from the expression

$$p(x,o) = \rho c V_n \tag{24}$$

where V_n is the particle velocity normal to the wavefront propagation direction. Therefore the relation between the V_y velocity component and the y direction resisting reaction is given by

$$(F^a_i)_y \approx (p(x,o)\cdot\Delta A) = \underbrace{\rho c\Delta A\ \text{Sin}\psi}_{\substack{\text{damping}\\ \text{constant}\equiv C_d}}\ V_y \tag{25}$$

Thus, the bottom face boundary condition (simulating an approximate wave absorbing boundary condition) is achieved by placing viscous dampers along the bottom cut (see figure 4b), wherein the damper constants are defined by

$$C_d = \rho c\Delta A\ \text{Sin}\psi \tag{26}$$

where ΔA is an appropriate area factor relating the pressure and concentrated force $(F^a_i)_y$. When the wave length of the incident wave is long relative to the mesh size, one can set ΔA = the surface element length for bottom surface nodes off the cuts and set ΔA = 1/2 the surface element length for the nodes lying on the cuts. Thus for the demonstration problem at hand, $C_d = 8.14587$ for the middle damper and half that amount for the end cut dampers.

Preparation of Demo NASTRAN Input Data

• executive control

The form of the DMAP instructions presented in Appendix A are general and are not problem dependent. The only times the user deviates from the presented DMAP sequence is (1) when he switches from one level of NASTRAN to another wherein the ALTER statement numbers change; or (2) when he wishes to turn off or turn on the intermediate matrix printout switch ISW, defined in the third DMAP ALTER statement (ISW = +1 prints all intermediate matrix operation steps in addition to a printout of the FORVEC vector which lists the left cut, interior, and right cut nodal forces; ISW = -1 no intermediate printout).

• case control

The standard CASE CONTROL deck is shown in the APPENDIX A, wherein the only thing worth noting is the fact that a DLOAD card is used for the purpose of superimposing the two vectors $\{F_\ell^a\}$ and $e^{-\mu^* i}$ $\{F_r^a\}$ which are both applied to the same left cut nodes (in correspondence with the first partition of Equation (10)).

• bulk data

The CDAMP2 cards are used to define the damping constants applied at the bottom surface nodes. The CQDMEM membrane elements (and corresponding MAT2, PQDMEM cards) are used to define the fluid media, employing the displacement fluid element approach described in ref. 8 . The collection of DLOAD, DAREA, DPHASE, RLOAD1, TABLED1 cards are used to insert the applied forces defined by Equations (22) and (23). The GRDSET card is employed to eliminate the non-applicable 3, 4, 5, 6 degrees-of-freedom for the 2-D membrane elements. Standard GRID cards define node coordinates and the standard FREQ card defines frequency of $f = \omega/2\pi = 3000$ Hz. The special nine PARAM cards defined by Equations (12) are evaluated using the data in Equations(19). In the case of the DMI cards for the partitioning vectors CV100, CV010, CV001, the first thing to establish for the problem is the sizes L, R, I for the individual partitions. It is convenient, for bookkeeping purposes, to number the left cut sequentially; and to also number the interior sequentially (with the lowest interior node number appearing next to the highest left cut node number) and finally number the right cut sequentially with the lowest right cut node appearing next to the highest internal node number. Thus for the problem at hand[†], L = 2 degrees-of-freedom/node times 3 left cut nodes = 6; I = 2 degrees-of-freedom/node times 3 interior nodes = 6; and R ≡ L = 6 due to the periodicity of the structure. The first 6 entries of CV100 are 1.0, the rest of the entries are zero; the 7th through 12th entries of CV010 are 1.0, the rest being zero; and finally the 13th through 18th entries of CV001 are 1.0, all other entries being zero. The 8 merging DMI cards LIXLI ••• L2XL2 are defined according to the instructions given earlier in the paper and need no further comment. Finally the unit matrix DMI cards for the UMATR matrix are defined according to the instructions given earlier in the paper.

Results of NASTRAN Demo Problem

Selected output results for the figure 4b demonstration problem are presented as NASTRAN direct printout (real and imaginary part type output format) in APPENDIX B. The first part of the NASTRAN printout illustrates the net input vector (e.g., results of Equations (22) and (23)); it is always good to include as a checking feature of the input.

Next the stress output is printed and refers to the stress computed at the center of the element. Note that due to the sign convention difference regarding stress and pressure (opposite in sign, see ref. 8 for details), the user must reverse the sign of stress to obtain the pressure, i.e.,

$$p = -\sigma_{xx} = -\sigma_{yy}$$

It should be noted that the NASTRAN output format headings are in error (due to a formating bug in NASTRAN that has remained in practically all levels of NASTRAN); the output heading should be read as follows (for each element row of stress output): Real Pt. (σ_{xx}), Real Pt. (σ_{yy}), Real Pt. (σ_{xy}), Img. Pt. (σ_{xx}), Img. Pt. (σ_{yy}), Img. Pt. (σ_{xy}). The results shown in APPENDIX B, Table B-1, show the NASTRAN results next to the exact solution, after converting the real and imaginary parts into amplitude and phase data.

The exact solution is evaluated with p(x,y) from Equation (14) at distances r that locate a line (drawn parallel to the wave front) which passes through the midpoint of an element. The NASTRAN results agree quite well with the exact solution in both phase and amplitude. It is noted that a very slight asymmetry exists between the NASTRAN phase angle results for element 3 and element 4. This is probably due to the fact that the boundary conditions on the top and right cut surfaces do not result in exactly the same applied nodal forces (e.g., the top forces were not computed using a consistent pressure-to-force application formula that involves the displacement shape functions). A similar comment applies to the relation between the bottom surface forces and left cut forces. When the program is run with the detailed print switch triggered (ISW = +1), the left and right cut forces are computed from the solution displacement vector and subsequently printed in condensed format (under the FORVEC header). Inspection of these results showed that corresponding left and right cut internal nodal forces differed by the proper amount, $\mu^* = 25.455^\circ$.

[†] Suppose, for illustrative purposes, node 5 had the x displacement SPC constrained to zero and suppose the x displacement of node 4 was forced (through an MPC relation) to equal the x displacement of node 6. In this particular case L = 2•3 = 6; I = 2•3-1-1 = 4; R ≡ L = 6.

CONCLUDING REMARKS

The DMAP sequence presented in APPENDIX A, permits the NASTRAN user to solve a class steady state dynamically loaded periodic structures, wherein the internal forces and resulting response at the periodic ends of the typical repeating substructure are related by a known complex phase relation, $e^{i\mu^*}$. The DMAP sequence is general and need not be changed from one problem to the next. When using the methodology presented here, one should be extremely careful that the CV100, CV010, CV001 partitioning vectors are prepared properly. For large problems, a preprocessor should be written which automatically generates these vectors. One should also be careful to allow for SPC, MPC and OMIT cards which compact the solution vector and consequently should be taken into account in the preparation of the partitioning vectors. The user is strongly advised to study the sample demonstration problem presented here before undertaking anymore complicated problems.

At this point, the DMAP sequence has been checked out for a set of small size problems, therefore comments regarding computer run time on large problems cannot be made at this time. Some of the DMAP operations can be increased in efficiency by employing the matrix operation ADD5 DMAP module in place of repeatedly employing the matrix ADD DMAP module. The ADD5 routine was not used in the 1108, level 15.5 version of NASTRAN due to the fact that the system did not consistently successfully add a string of 5 matrices during certain checkout phases of the programming.

Finally, it is advised that before attempting to exercise the DMAP sequence presented in APPENDIX A, one should run the demonstration problem as a bench mark check in. If a Univac 1108, level 15.5 version is used, it may be necessary to increase the "max-files" to 40 in NTRAN$.

REFERENCES

1. Brillouin, L., "Wave Propagation in Periodic Structures", Dover Publications, Inc., New York, 1946.

2. Elachi, C., "Waves in Active and Passive Periodic Structures", Proceedings of the IEEE, Vol. 64, No. 12, Dec. 1976.

3. Heckl, M. A., "Investigations on the Vibrations of Grillages and Other Simple Beam Structures, Journal of the Acoustical Society of America, Vol. 36, 1964.

4. Abrahamson, A. L., "Flexual Wave Mechanics - an Analytical Approach to the Vibration of Periodic Structures Forced by Convected Pressure Fields", Journal of Sound and Vibration, Vol. 28, 1973.

5. Murakami, H. and Luco, E., "Seismic Response of a Periodic Array of Structures", Journal of the Engineering Mechanics Div., ASCE, Oct. 1977.

6. Mead, D. J., "A General Theory of Harmonic Wave Propagation in Linear Periodic Systems with Multiple Coupling", Journal of Sound and Vibration, Vol. 27, No. 2, Feb. 1973.

7. Orris, R. M. and Petyt, M., "A Finite Element Study of Harmonic Wave Propagation in Periodic Structures", Journal of Sound and Vibration, Vol. 33, No. (2), Feb. 1974.

8. Kalinowski, A. J. and Patel, J. S., "A Summary of NASTRAN Fluid/Structure Interaction Capabilities, NASTRAN Users' Experiences", Fifth Colloquium, NASA TM X-3428, Oct. 1976.

APPENDIX-A

(NASTRAN DEMONSTRATION PROBLEM INPUT)

NASTRAN EXECUTIVE CONTROL DECK

```
ID NUSC PERIODIC STRUCTURE SAMPLE
APP DISP
SOL 8,0
TIME 30
DIAG 2,3,8,14,15
DIAG 22
ALTER 138

$ DMPA INSTRUCTIONS FOR PERIODIC STRUCTURES (CODED BY A.J.KALINOWSKI)
$ ISW=+1 PRINT  ... =-1 NO PRINT    OF ALL KEY MATRICES FOR PERIODIC STR.
PARAM //C,N,NOP/V,N,ISW=+1 $
PARAM //C,N,NOP/V,N,IM1=-1 $
PARAM //C,N,NOP/V,N,PURSW1=-1 $
$ DEFINE ROW PARTITIONING VECTORS WITH DUMMY ADD OPERATION
$  WHERE CV100 CV010 CV001                ARE READ IN ON DMI BULK DATA
ADD CV100,/RV100 $
ADD CV010,/RV010 $
ADD CV001,/RV001 $
COND MAR,ISW $
MATPRN RV100,RV010,RV001,,// $
MATPRN CV100,CV010,CV001,,// $
LABEL MAR $
$ CONDITIONAL PRINT OF KDD,MDD, BDD BEFORE PARTITION APPLIED (
COND KAL,ISW $
MATPRN KDD,MDD,BDD,,// $
$  SAVE ORIGINAL KDD MDD BDD MATRICES FOR PROCESSING FORCES AT CUTS
ADD KDD,/SAVKDD $
ADD MDD,/SAVMDD $
ADD BDD,/SAVBDD $
LABEL KAL$
$ PARITION K,M,B MATRICES
$ PARTITION LL BLOCK
PARTN KDD,CV100,RV100/,,,DKLL/C,N,1 $
PARTN MDD,CV100,RV100/,,,DMLL/C,N,1 $
PARTN BDD,CV100,RV100/,,,DBLL/C,N,1 $
$ PARTITION LI BLOCK
PARTN KDD,CV010,RV100/,,,DKLI/C,N,1 $
PARTN MDD,CV010,RV100/,,,DMLI/C,N,1 $
PARTN BDD,CV010,RV100/,,,DBLI/C,N,1 $
$ PARTITION LR BLOCK
PARTN   KDD,CV001,RV100/,,,DKLR/C,N,1 $
PARTN   MDD,CV001,RV100/,,,DMLR/C,N,1 $
PARTN   BDD,CV001,RV100/,,,DBLR/C,N,1 $
$ PARTITION IL BLOCK
PARTN   BDD,CV100,RV010/,,,DBIL/C,N,1 $
PARTN   KDD,CV100,RV010/,,,DKIL/C,N,1 $
PARTN   MDD,CV100,RV010/,,,DMIL/C,N,1 $
$ PARTITION II BLOCK
PARTN   KDD,CV010,RV010/,,,DKII/C,N,1 $
PARTN   MDD,CV010,RV010/,,,DMII/C,N,1 $
PARTN   BDD,CV010,RV010/,,,DBII/C,N,1 $
$ PARTITION IR BLOCK
PARTN   KDD,CV001,RV010/,,,DKIR/C,N,1 $
PARTN   MDD,CV001,RV010/,,,DMIR/C,N,1 $
PARTN   BDD,CV001,RV010/,,,DBIR/C,N,1 $
$ PARTITION RL BLOCK
PARTN   KDD, CV100, RV001/,,,DKRL/C,N,1 $
PARTN   MDD, CV100, RV001/,,,DMRL/C,N,1 $
PARTN   BDD, CV100, RV001/,,,DBRL/C,N,1 $
$ PARTITION RI BLOCK
PARTN   KDD,CV010,RV001/,,,DKRI/C,N,1 $
PARTN   MDD,CV010,RV001/,,,DMRI/C,N,1 $
PARTN   BDD,CV010,RV001/,,,DBRI/C,N,1 $
$ PARTITION RR BLOCK
PARTN   KDD, CV001, RV001/,,,DKRR/C,N,1 $
PARTN   MDD, CV001, RV001/,,,DMRR/C,N,1 $
PARTN   BDD, CV001, RV001/,,,DBRR/C,N,1 $
$CONDITIONAL PRINT OF PARTITIONS
COND KALIN,ISW $
MATPRN DKLL,DMLL,DBLL,,// $
MATPRN DKLI,DMLI,DBLI,,// $
MATPRN DKLR,DMLR,DBLR,,// $
MATPRN DKIL,DMIL,DBIL,,// $
MATPRN DKII,DMII,DBII,,// $
MATPRN DKIR,DMIR,DBIR,,// $
MATPRN DKRL,DMRL,DBRL,,// $
MATPRN DKRI,DMRI,DBRI,,// $
MATPRN DKRR,DMRR,DBRR,,// $
LABEL KALIN $
$   *
$ FORM  PARTITIONS OF ASSEMBLED MATRIX  MDDX, BDDX, KDDX
$   *
$ FORM LL BLOCK
$
$                       FORM AMLL
ADD DMLL,DMLR/D1AMLL/V,Y,PONE/V,Y,CMST $
ADD D1AMLL,DMRL/D2AMLL/V,Y,PONE/V,Y,CMST $
ADD D2AMLL,DMRR/AMLL/V,Y,PONE/V,Y,PONE $
PURGE KDD,MDD,BDD/PURSW1 $
PURGE D1AMLL,D2AMLL/PURSW1 $
```

```
$                     FORM ABLL
ADD DBLL,DBLR/D1BLL/V,Y,PONE/V,Y,CMST $
ADD D1BLL,DMLR/D2BLL/V,Y,PONE/V,Y,NSMSTO $
ADD D2BLL,DKLR/D3BLL/V,Y,PONE/V,Y,SMSBO $
ADD D3BLL,DBRR/D4BLL/V,Y,PONE/V,Y,PONE  $
PURGE D1ABLL,D2ABLL,D3ABLL/PURSW1 $
ADD D4BLL,DMRL/D5BLL/V,Y,PONE/V,Y,SMSTO $
ADD D5BLL,DKRL/ABLL/V,Y,PONE/V,Y,NSMSBO $
PURGE D4BLL,D5BLL/PURSW1 $
$                     FORM AKLL
ADD DKLL,DKLR/D1KLL/V,Y,PONE/V,Y,CMST $
ADD D1KLL,DBLR/D2KLL/V,Y,PONE/V,Y,NSMSTO $
ADD D2KLL,DKRL/D3KLL/V,Y,PONE/V,Y,CMST $
ADD D3KLL,DKRR/D4KLL/V,Y,PONE/V,Y,PONE $
ADD D4KLL,DBRL/AKLL/V,Y,PONE/V,Y,SMSTO $
PURGE D1KLL,D2KLL,D3KLL,D4KLL/PURSW1 $
$ FORM LI BLOCK
$
$                     FORM AMLI
ADD DMLI,DMRI/AMLI/V,Y,PONE/V,Y,CMST $
$                     FORM ABLI
ADD DBLI,DBRI/D1BLI/V,Y,PONE/V,Y,CMST $
ADD D1BLI,DMRI/D2BLI/V,Y,PONE/V,Y,SMSTO $
ADD D2BLI,DKRI/ABLI/V,Y,PONE/V,Y,NSMSBO $
PURGE D1BLI,D2BLI/PURSW1 $
$                     FORM AKLI
ADD DKLI,DKRI/D1KLI/V,Y,PONE/V,Y,CMST $
ADD D1KLI,DBRI/AKLI/V,Y,PONE/V,Y,SMSTO $
PURGE D1KLI/PURSW1 $
$ FORM IL BLOCK
$
$                     FORM AMIL
ADD DMIL,DMIR/AMIL/V,Y,PONE/V,Y,CMST $
$                     FORM ABIL
ADD DBIL,DBIR/D1BIL/V,Y,PONE/V,Y,CMST $
ADD D1BIL,DMIR/D2BIL/V,Y,PONE/V,Y,NSMSTO $
ADD D2BIL,DKIR/ABIL/V,Y,PONE/V,Y,SMSBO $
PURGE D1BIL,D2BIL/PURSW1 $
$                     FORM AKIL
ADD DKIL,DKIR/D1KIL/V,Y,PONE/V,Y,CMST $
ADD D1KIL,DBIR/AKIL/V,Y,PONE/V,Y,NSMSTO $
PURGE D1KIL/PURSW1 $
$  FORM  II BLOCK
$   AMII SAME AS DMII   ABII SAME AS DMII   AKII SAME AS DKII
$  FORM RL BLOCK
```

```
ADD UMATR,/ABRL/V,Y,SMSBO $
ADD UMATR,/AKRL/V,Y,CMST $
$ FORM RR BLOCK
ADD UMATR,/AKRR/V,Y,NONE $
$  THE   LR , IR ,RI BLOCKS ARE NULL
$  PRINT BLOCKS BEFORE ASSEMBLY
COND DEB,ISW $
MATPRN AMLL,ABLL,AKLL,,// $
MATPRN AMLI,ABLI,AKLI,,// $
MATPRN AMIL,ABIL,AKIL,,// $
MATPRN ABRL,AKRL,AKRR,,// $
PRTPARM //C,N,0$
LABEL DEBS
$   *
$ NEXT ASSEMBLE BLOCKS WITH MERGE
$   *
$      MERGE  LL BLOCK
MERGE LIXLI,,,AKLL,CV100,RV100/KDDXLL/C,N,1 $
MERGE LIXLI,,,AMLL,CV100,RV100/MDDXLL/C,N,1 $
MERGE LIXLI,,,ABLL,CV100,RV100/BDDXLL/C,N,1 $
$  MERGE LI BLOCK
MERGE LIXL2,,,AKLI,CV010,RV100/KDDXLI/C,N,1 $
MERGE LIXL2,,,AMLI,CV010,RV100/MDDXLI/C,N,1 $
MERGE LIXL2,,,ABLI,CV010,RV100/BDDXLI/C,N,1 $
$  FORM PARTIAL SUM
ADD KDDXLL,KDDXLI/SUMK1 $
ADD MDDXLL,MDDXLI/SUMM1 $
ADD BDDXLL,BDDXLI/SUMB1 $
$   PURGE COMPONENTS OF PARTIAL SUM NOT NEEDED ANY MORE
PURGE KDDXLL,KDDXLI,MDDXLL,MDDXLI,BDDXLL,BDDXLI/PURSW1 $
$  MERGE IL BLOCK
MERGE L2XLI,,,AKIL,CV100,RV010/KDDXIL/C,N,1 $
MERGE L2XLI,,,AMIL,CV100,RV010/MDDXIL/C,N,1 $
MERGE L2XLI,,,ABIL,CV100,RV010/BDDXIL/C,N,1 $
$   CONTINUE PARTIAL SUM
ADD SUMK1,KDDXIL/SUMK2 $
ADD SUMM1,MDDXIL/SUMM2 $
ADD SUMB1,BDDXIL/SUMB2 $
COND JOHN,ISW $
MATPRN SUMK1,KDDXIL,SUMK2,SUMM1,MDDXIL//$
MATPRN SUMM2,SUMB1,BDDXIL,SUMB2,//$
LABEL JOHN $
$   PURGE COMPONENTS OF PARTIAL SUM NOT NEEDED ANY MORE
PURGE SUMK1,KDDXIL,SUMM1,MDDXIL,SUMB1,BDDXIL/PURSW1 $
$  MERGE  II BLOCK
```

```
MERGE L2XL2,,,DKII,CV010,RV010/KDDXII/C,N,1 $
MERGE L2XL2,,,DMII,CV010,RV010/MDDXII/C,N,1 $
MERGE L2XL2,,,DBII,CV010,RV010/BDDXII/C,N,1 $
$   CONTINUE PARTIAL SUM
ADD SUMK2,KDDXII/SUMK3 $
ADD SUMM2,MDDXII/MDDX $
$   WHERE MDDX IS THE FINAL MASS MATRIX SUM
ADD SUMB2,BDDXII/SUMB3 $
COND BULL,ISW $
MATPRN SUMK2,KDDXII,SUMK3,SUMM2,MDDXII//$
MATPRN MDDX,SUMB2,BDDXII,SUMB3,//$
LABEL BULL$
$   PURGE COMPONENTS OF PARTIAL SUM NOT NEEDED ANY MORE
PURGE SUMK2,KDDXII,SUMM2,MDDXII,SUMB2,BDDXII/PURSW1 $
$     MERGE RL BLOCK
MERGE LIXLI,,,AKRL,CV100,RV001/KDDXRL/C,N,1 $
MERGE LIXLI,,,ABRL,CV100,RV001/BDDXRL/C,N,1 $
$   CONTINUE PARTIAL SUM
ADD SUMK3,KDDXRL/SUMK4 $
ADD SUMB3,BDDXRL/BDDX $
$   WHERE BDDX IS THE FINAL MATRIX SUM
COND PILL,ISW $
MATPRN SUMK3,KDDXRL,SUMK4,SUMB3,BDDXRL//$
MATPRN BDDX,,,,//$
LABEL PILL$
$   PURGE COMPONENTS OF PARTIAL SUM NOT NEEDED ANY MORE
PURGE SUMK3,KDDXRL,SUMB3,BDDXRL/PURSW1 $
$     MERGE RR BLOCK
MERGE LIXLI,,,AKRR,CV001,RV001/KDDXRR/C,N,1 $
$   CONTINUE PARTIAL SUM
ADD SUMK4,KDDXRR/KDDX $
$   WHERE KDDX IS THE FINAL SUMMED MATRIX
COND BOOK,ISW $
MATPRN SUMK4,KDDXRR,KDDX,,//$
LABEL BOOK $
$   PURGE COMPONENTS OF PARTIAL SUM NOT NEEDED ANY MORE
PURGE SUMK4,KDDXRR/PURSW1 $
$   *
$    *** NOW ALL THE KDDX   BDDX  AND MDDX MATRICES ARE FORMED
$   *
COND JACK,ISW $
MATPRN MDDX,BDDX,KDDX,,// $
LABEL JACK$
ALTER 139,139
FRRD CASEXX,USE[illegible],DLT,FRL,GMD,GOD,KDDX,BDDX,MDDX,,DIT/UDVF,PSF,PDF,PPF/
```

```
      C,N,DISP/C,N,DIRECT/V,N,LUSETD/V,N,MPCF1/V,N,SINGLE/V,N,OMIT/
      V,N,NONCUP/V,N,FRQSET/C,Y,DECOMOPT=1 $
ALTER 140
COND ABE,ISW $
$ PRINT THE FORCE VECTORS USED BY THE EQUIVALENT PERIODIC STRUCTURE
MATPRN PDF,PSF,PPF,,// $
$ PROCESS THE LEFT AND RIGHT PERIODIC CUT FORCES
ADD SAVMDC,SAVKDD/DSAV/V,Y,NOMEG2/V,Y,PONE $
ADD DSAV,SAVBDD/KNET/V,Y,PONE/V,Y,FIOMFG $
PURGE DSAV/PURSW1 $
MPYAD KNET,UDVF,/FORVEC/C,N,0/C,N,1/C,N,0/C,N,1 $
MATPRN KNET,UDVF,FORVEC,,// $
LABEL ABE $
ENDALTER

CEND
```

```
                    C A S E   C O N T R O L   D E C K   E C H O

TITLE = PERIODIC STRUCTURE (FIRST SAMPLE)
MAXLINES = 30000
DLOAD = 10
FREQUENCY = 1
OLOAD = ALL
STRESS = ALL
DISPLACEMENT = ALL
BEGIN BULK
```

```
                              S O R T E D   B U L K   D A T A   E C H O

.   1  ..   2  ..   3  ..   4  ..   5  ..   6  ..   7  ..   8  ..   9  ..  10  .
CDAMP2  101     4.07293 1       2
CDAMP2  106     8.14587 6       2
CDAMP2  107     4.07293 7       2
CQDMEM  1       1       2       5       4       3
CQDMEM  2       1       1       6       5       2
CQDMEM  3       1       4       5       8       9
CQDMEM  4       1       5       6       7       8
DAREA   21      3       2       -49.8972
DAREA   21      4       2       -99.7944
DAREA   210     3       2       -49.9742
DLOAD   10      1.      1.      100     1.      101
DMI     CV001   0       2       1       1               18      1
DMI     CV001   1       13      1.      1.      1.      1.      1.      DOGS
+OGS    1.
DMI     CV010   0       2       1       1               18      1
DMI     CV010   1       7       1.      1.      1.      1.      1.      PIGS
+IGS    1.
DMI     CV100   0       2       1       1               18      1
DMI     CV100   1       1       1.      1.      1.      1.      1.      CATS
+ATS    1.
DMI     LIXLI   0       2       1       1               12      12
DMI     LIXLI   1       1       .0
DMI     LIXL2   0       2       1       1               12      12
DMI     LIXL2   1       1       .0
DMI     L2XLI   0       2       1       1               12      12
DMI     L2XLI   1       1       .0
DMI     L2XL2   0       2       1       1               12      12
DMI     L2XL2   1       1       .0
DMI     UMATR   0       6       1       2               6       6
DMI     UMATR   1       1       1.
DMI     UMATR   2       2       1.
DMI     UMATR   3       3       1.
DMI     UMATR   4       4       1.
DMI     UMATR   5       5       1.
DMI     UMATR   6       6       1.
DPHASE  23      3       2       28.6378
DPHASE  23      4       2       38.18377
DPHASE  230     3       2       22.27387
FREQ    1       3000.
GRDSET                                                  3456
GRID    1               .0      .0      .0
GRID    2               .0      1.0     .0
GRID    3               .0      2.0     .0
GRID    4               1.0     2.0     .0
GRID    5               1.0     1.0     .0
GRID    6               1.0     .0      .0
GRID    7               2.0     .0      .0
GRID    8               2.0     1.0     .0
GRID    9               2.0     2.0     .0
MAT2    10      345600. 345600. .0      345600.                 .000096
PARAM   CMST    .9029168.0
PARAM   FIOMEG  1.0-20  18849.55
PARAM   NOMEG2  -3.553+8.0
PARAM   NONE    -1.0    .0
PARAM   NSMSBO  -2.280-5.0
PARAM   NSMSTO  -8108.830.0
PARAM   PONE    +1.0    .0
PARAM   SMSBO   +2.280-5.0
PARAM   SMSTO   +8108.830.0
PQDMEM  1       10      1.0
RLOAD1  100     21              23      22
RLOAD1  101     210             230     22
TABLED1 22                                                              PULS1
+ULS1   0.0     1.0     100000. 1.0     ENDT
ENDDATA
```

```
FREQUENCY =   3.000000+03
                                   C O M P L E X   L O A D   V E C T O R
                                                (REAL/IMAGINARY)

POINT ID.   TYPE          T1                  T2              T3               R1               R2               R3
      3       G           .0              -9.003838+01    .0               .0               .0               .0
                          .0              -4.285621+01    .0               .0               .0               .0

      4       G           .0              -7.844159+01    .0               .0               .0               .0
                          .0              -6.169147+01    .0               .0               .0               .0

FREQUENCY =   3.000000+03
         C O M P L E X   S T R E S S E S   I N   Q U A D R I L A T E R A L   M E M B R A N E S   ( C Q D M E M )
                                                ~~(REAL/IMAGINARY)~~

   ELEMENT                                        - STRESSES IN ELEMENT COORDINATE SYSTEM -
     ID.          real σxx ~~NORMAL-X~~  real σyy        real σxy    ~~NORMAL-Y~~  imag σxx            imag σyy  ~~SHEAR-XY~~ imag σxy
        1      -9.069812+01/,   -9.069812+01         .0            /,   -4.316528+01          -4.316528+01/,    .0
        2      -9.808248+01/,   -9.808248+01         .0            /,   -2.207309+01          -2.207309+01/,    .0
        3      -7.895863+01/,   -7.895863+01         .0            /,   -6.209007+01          -6.209007+01/,    .0
        4      -9.080797+01/,   -9.080797+01         .0            /,   -4.314283+01          -4.314283+01/,    .0
```

TABLE B-1

NASTRAN-EXACT SOLUTION COMPARISON

p / element	NASTRAN |p|, psi	EXACT |p|, psi	NASTRAN Phase of p	EXACT Phase of p
1	100.53	100.00	12.683°	12.728°
2	100.44	100.00	25.451°	25.456°
3	100.44	100.00	38.180°	38.183°
4	100.53	100.00	25.412°	25.456°

MICROFILMED FROM
BEST AVAILABLE COPY

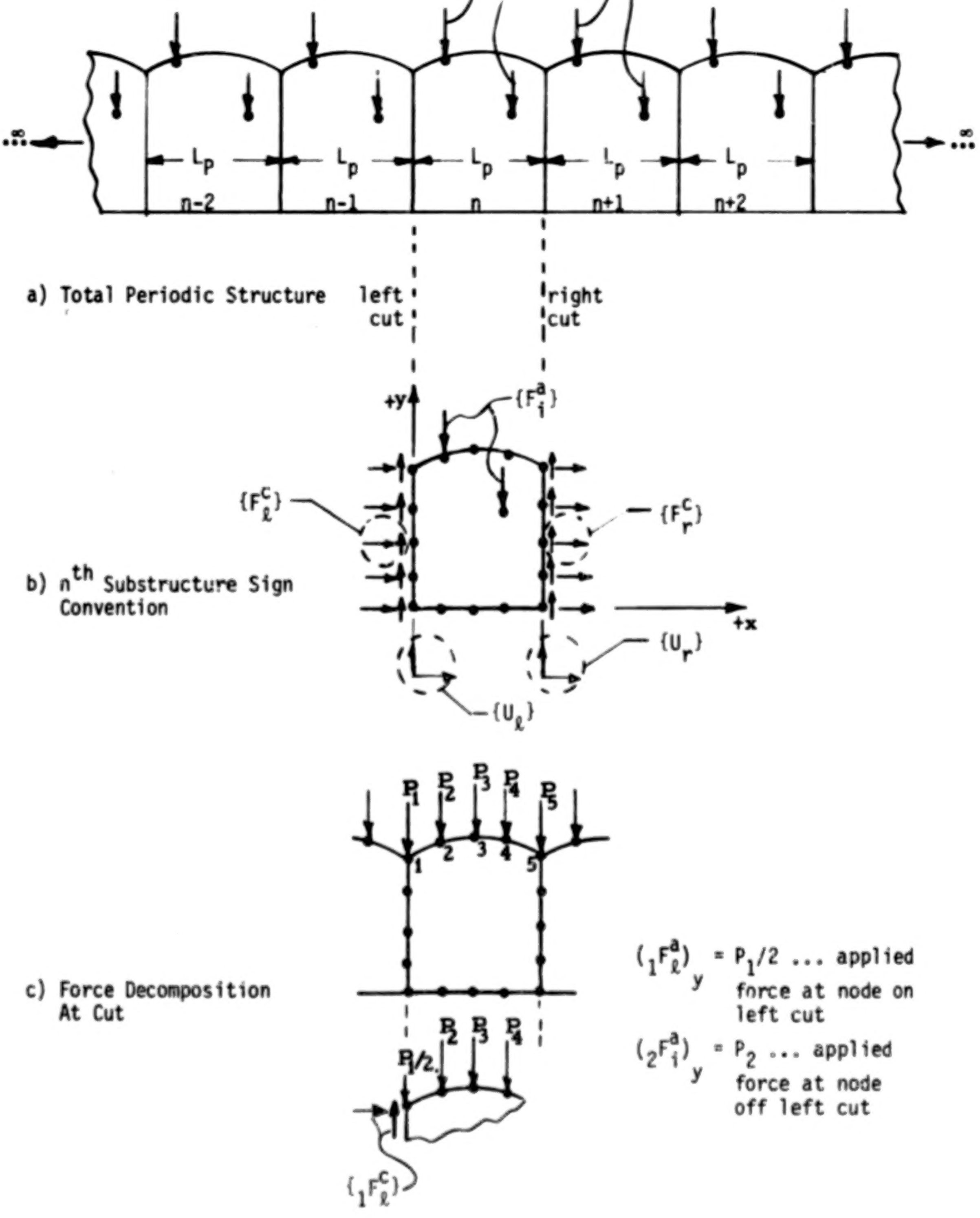
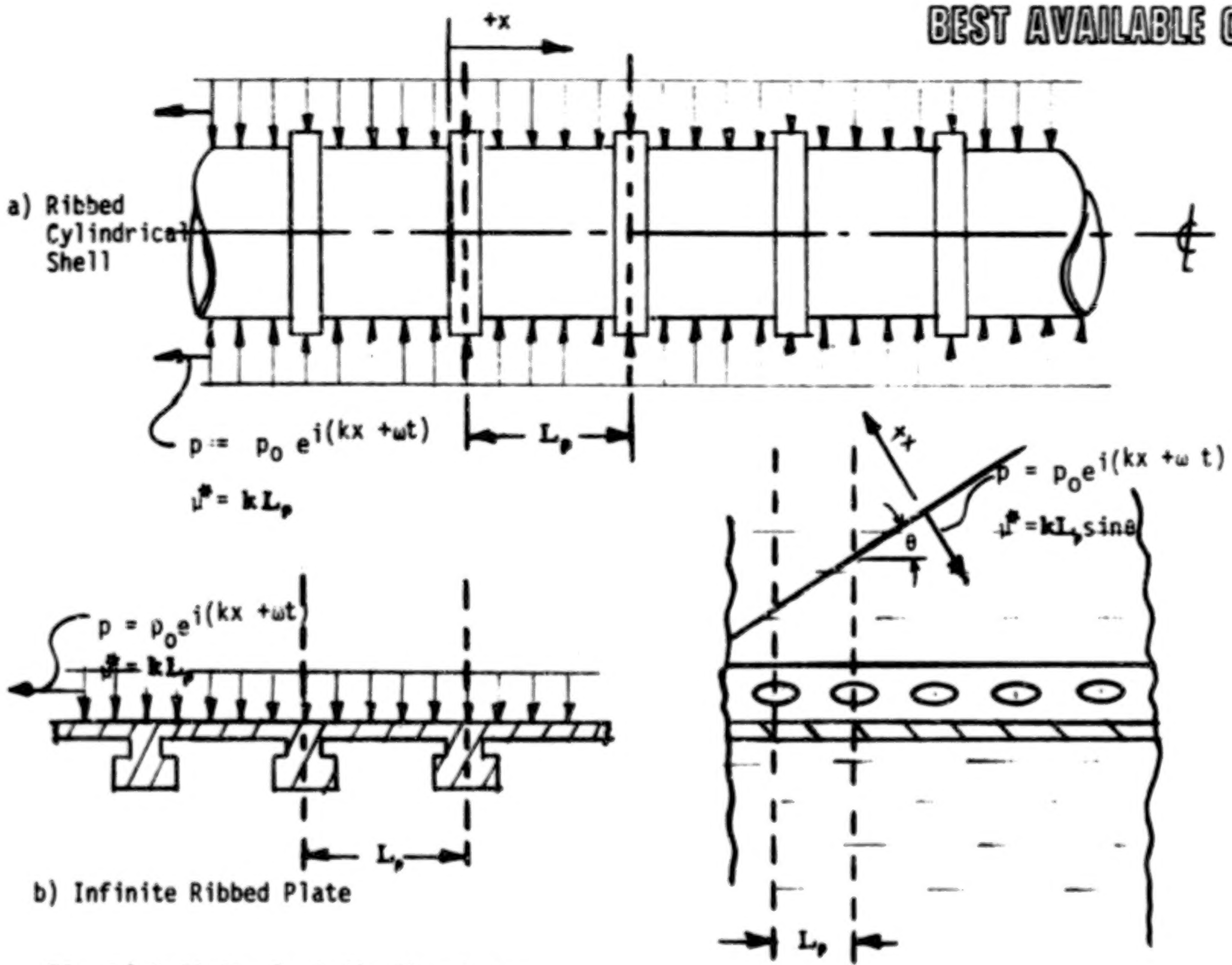

{F}_n
{F}_{n+1}
∞
∞
L_p
L_p
L_p
L_p
L_p
n-2
n-1
n
n+1
n+2
a) Total Periodic Structure
left cut
right cut
+y
{F_i^a}
{F_ℓ^c}
{F_r^c}
b) n^th Substructure Sign Convention
+x
{U_r}
{U_ℓ}
P1 P2 P3 P4 P5
1 2 3 4 5
c) Force Decomposition At Cut
P2 P3 P4
P1/2
{1F_ℓ^c}
(1F_ℓ^a)_y = P1/2 ... applied force at node on left cut
(2F_i^a)_y = P2 ... applied force at node off left cut

Fig. 1 Periodic Structure Notation

MICROFILMED FROM
BEST AVAILABLE COPY


+x
$p = p_o e^{i(kx + \omega t)}$
$\mu^* = kL_p$
L_p
x_+
$p = p_o e^{i(kx + \omega t)}$
$\mu^* = kL_p \sin\theta$
θ
L_p
$p = p_o e^{i(kx + \omega t)}$
$\mu^* = kL_p$
L_p
℄

a) Ribbed Cylindrical Shell

b) Infinite Ribbed Plate

Fig. 2) Infinite Periodic Structures

c) Submerged Layered Plate

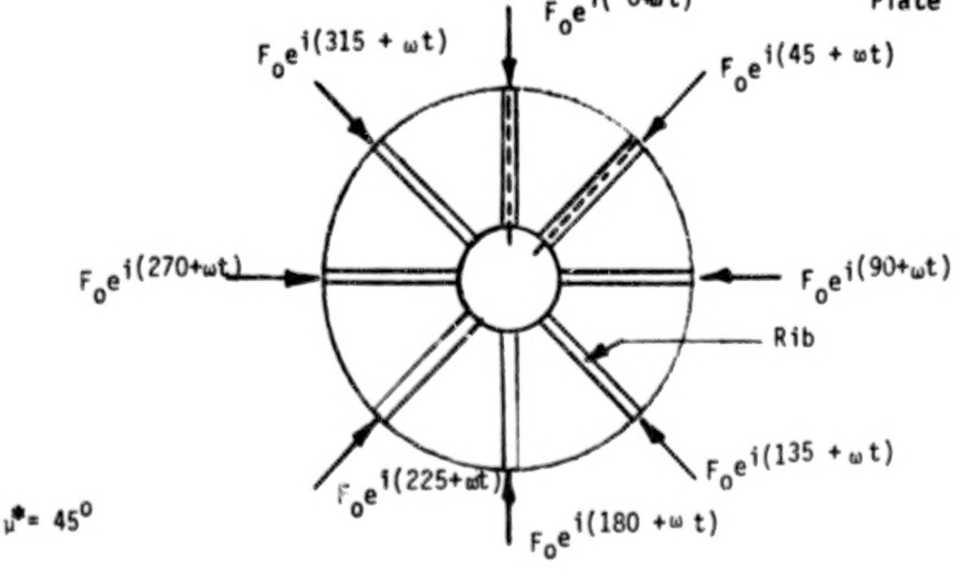
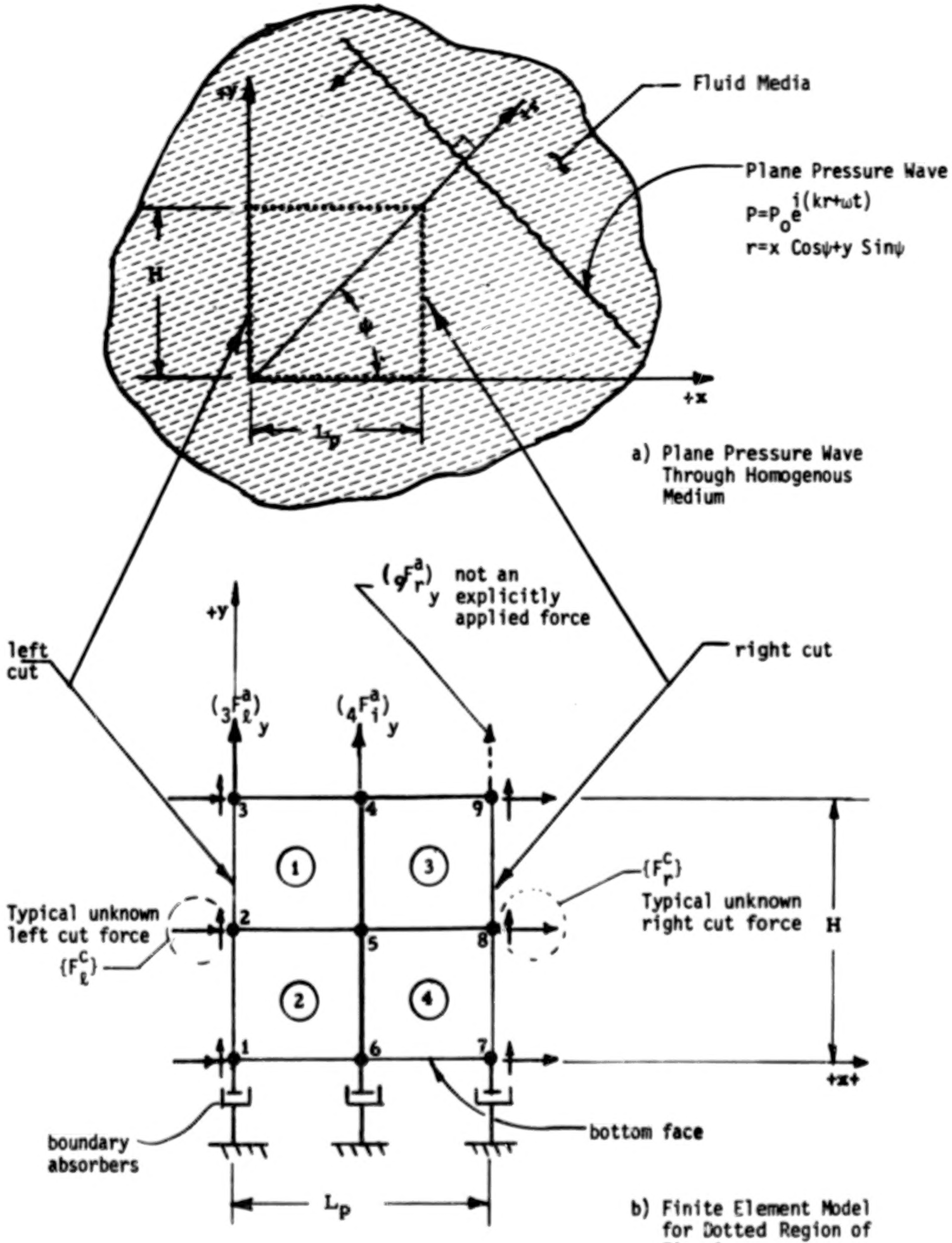

$F_o e^{i(0+\omega t)}$
$F_o e^{i(315 + \omega t)}$
$F_o e^{i(45 + \omega t)}$
$F_o e^{i(270+\omega t)}$
$F_o e^{i(90+\omega t)}$
Rib
$F_o e^{i(225+\omega t)}$
$F_o e^{i(135 + \omega t)}$
$F_o e^{i(180 + \omega t)}$
$\mu^* = 45^0$

Fig. 3 Cyclic Periodic Structure


Fluid Media
Plane Pressure Wave
$P=P_o e^{i(kr+\omega t)}$
$r=x \cos\psi + y \sin\psi$
+y
+r
H
ψ
+x
L_p
$({}_9F^a_r)_y$ not an explicitly applied force
+y
left cut
right cut
$({}_3F^a_\ell)_y$
$({}_4F^a_i)_y$
1
2
3
4
5
6
7
8
9
①
②
③
④
Typical unknown left cut force
$\{F^c_\ell\}$
$\{F^c_r\}$
Typical unknown right cut force
H
+x+
bottom face
boundary absorbers
L_p

a) Plane Pressure Wave Through Homogenous Medium

b) Finite Element Model for Dotted Region of Fig. 4a

Fig. 4 Demonstration Problem

APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS AT NORTHROP

Ashok K. Singh

SUMMARY

A comprehensive gust response analysis has been performed on a complete model of an airplane using the NASTRAN aeroelastic package by the Advanced Structural Computer Methods (ASCM) group at Northrop. Earlier the same model was used to perform subsonic flutter analysis of the airplane using the computer program. Both the random and discrete gust response analyses have been performed including the control system dynamics in the problem. On a large aircraft gust response analysis including the flexible modes of the vehicles is a major design task. On a light weight fighter aircraft the analysis is primarily performed in order to study the ride quality and to provide the frequency of exceedance curves for the control surface hinge moments and some selected dynamic loads for static and fatigue analysis.

INTRODUCTION

The NASTRAN finite element program has been used at Northrop since 1972. The ASCM group has been actively evaluating and exercising the various NASTRAN dynamics analysis features and the aeroelastic package for several months. Integrated NASTRAN structural analysis combining static and dynamic analysis, e.g., flutter and gust, is the planned goal at Northrop. In order to achieve this end, a common structural model throughout an engineering project must be used. This practice is also expected to minimize the use of inconsistent structural data and unnecessary data handling among the various engineering disciplines.

Symmetric response analysis is evaluated in two applications to the complete airplane aeroelastic model. The analyses are random response to atmospheric turbulence and transient response to a discrete gust. The structural and aerodynamic models are the same as that used in the flutter analysis as presented in Reference 1.

The gust response analysis by NASTRAN can only be performed with the aerodynamic forces computed by the Doublet-Lattice Method. Supersonic gust response capability has not been provided. Spacewise variation of the gust velocity is not allowed in the present NASTRAN formulation but should be considered for future development. The gust velocity normal to the free stream velocity is taken as an additional source of downwash in the computation of the aerodynamic forces. The standard forms of power spectrum of the atmospheric turbulence available in NASTRAN are Von Karman and Dryden. However, provision is made to use any other form of spectrum by the means of tabular input. In the case of the random gust analysis, the frequency response, power spectral density, root mean square value and the frequency of zero crossing, N_o, of the response are output. A post processor can be easily written in order to compute the $\bar{A}$ and frequency of exceedance N(y) of the response. A procedure of weighting the aerodynamic forces to match measured data on each panel must be developed.

The discrete gust analysis is performed by the Fourier method. First, the time varying loads are transformed into the frequency domain by Fourier series or Fourier integrals. Second, the responses computed in the frequency domain are converted into the time domain by inverse Fourier transform methods. Three approximate methods are available in order to evaluate the inverse.

A flight control system is incorporated in the NASTRAN model that utilizes a Ride Improvement Mode System (RIMS) and pitch Control Augmentation System (CAS). A dedicated accelerometer and pitch rate gyro near the pilot station are chosen to measure acceleration and pitch rate which are fedback as the control signal in order to actuate the flaperon and stabilator (Figure 1).

To obtain satisfactory ride qualities during low altitude high speed flight, extensive effort has been expended in the development of a ride improvement system. The system tries to maintain a constant value of lift for changes in angle-of-attack due to turbulence. This is performed by sensing the load factor at the pilot station and using that as a control signal to command the high rate flaperons, so that the flaperons can minimize the turbulence-induced incremental load factor at the pilot station.

Operating in parallel with the ride mode is the CAS which uses a blend of load factor and pitch rate to maintain aircraft stability while trying to minimize uncommanded pitch rate and load factor.

With the advent of the control configured vehicle, today's control system engineer must have a thorough knowledge of aeroservoelastic behavior of the flying machine. The design of the filters in the feedback loop cannot be completed without the knowledge of elastic and aeroelastic characteristics of the modern aircaraft. A new engineering discipline of aeroservoelasticity is emerging which will play a prominent role in the early design phases of an integrated control system.

SYMBOLS

$\bar{A}$	Ratio of root-mean-square value of load to root-mean-square value of gust velocity
b_1 b_2	Intensity parameters in the expression for probability of σ_a
$H_{ja}(\omega)$	Frequency response due to the gust excitation
N_o	Average number of zero crossings with positive slope, per unit time

$N(y)$	Number of exceedances of the indicated value of y per unit time
P_1, P_2	Fractions of total flight time in non-storm and storm turbulence respectively
$S_a(\omega)$	Power spectral density of gust velocity
$S_j(\omega)$	Power spectral density of response
σ_r	Root-mean-square value of response quantity
σ_a	Root-mean-square value of gust velocity
ω	Circular frequency
ω_c	Cutoff frequency beyond which aeroelastic responses are no longer significant in turbulence

NASTRAN GUST RESPONSE ANALYSIS

The gust response analysis is performed on a complete aircraft in the following steps:

- Random response to Von Karman gust spectrum without control system interaction
- Random response to Von Karman gust spectrum with control system interaction
- Transient response to a discrete gust with control system interaction

RANDOM RESPONSE ANALYSIS

A NASTRAN beam element model of a complete airplane was used to perform the gust response analysis. The airplane with a tip store, launcher rail, wing, flaperon, fuselage, fin with rudder and horizontal stabilizer was modeled as finite beam elements as shown in Figure 2. The store and launcher rail assembly was tied to the wing tip by rigid elements which may be modified to possess elastic properties. The wing root and fin root flexibilities were modeled by lumped springs, which may be made more complex as the finite element model of the airplane is developed. The horizontal stabilizer root stiffness is a general element accounting for the spindle and the actuator assembly flexibilities. Mass properties were input on lumped mass element cards.

A doublet-lattice finite element program was used to represent the aerodynamics of the vehicle as shown in Figure 3. The wing with the launcher rail, fin with the rudder and horizontal stabilizer are represented as lifting surface elements. The fuselage is represented as slender body and interference elements. In the present analysis, the aerodynamic induction effect among all the elements is considered. The wing, horizontal stabilizer and fin are divided into 145, 20 and 20 micro lifting surface elements, respectively. The fuselage is divided into 14 slender body elements. There are 11 interference elements on the fuselage.

A complicated network of linear spline functions is used to relate the modal deflections to each of the aerodynamic element deflections. Five distinct splines were used for the wing, rail and flaperon panels, two for the horizontal stabilizer, three for the fin with rudder, two for the fuselage and one for the store.

The flight control system is represented in the model as 9 extra points and 2 scalar points with their coefficients in the mass, damping and stiffness matrices in order to represent the filters in the feedback loop. The two scalar points are the relative rotations of the flaperon and the stabilator. Constraint forces in the equations of motion due to the control laws are introduced by the Lagrange multiplier technique. An accelerometer and a pitch rate gyro are located near the pilot station in order to measure the airplane responses. The measured data are fedback in order to activate RIMS and CAS laws, which control the aircraft response.

The unsteady aerodynamics for Mach 0.8 and sea level is generated for eight reduced frequencies using the Doublet-Lattice Method available in NASTRAN. Symmetric flight condition is considered in the analysis. Three symmetric rigid body modes and twenty-two elastic modes are selected to generalize the aerodynamic forces. Only the modal method of aeroelastic response computation is available in the program. Generalized aerodynamic forces at other intermediate reduced frequencies are computed by means of a linear spline interpolation routine.

The gust response analysis of a light weight fighter considered in this problem is primarily performed in order to study the ride quality of the vehicle. In order to increase the survivability of modern fighter aircraft, new emphasis is being given to the capability of low-altitude high-speed penetration. For such an aircraft, the low-wing loading/high-lift curve slope which maximizes turn rate capability and maneuverability essential for survival in air combat also tends to deteriorate the ride quality during high-speed penetration. This can lead to reduced mission success in attacking heavily defended ground targets, or in the worst case even mission failure.

For the evaluation of ride quality through turbulence, $\bar{A}$ is used. $\bar{A}$ is the root-mean-square (rms) of the response divided by the rms gust level in feet per second (fps) as defined in Equation 1.

$$\bar{A} = \left[\frac{\int_0^{\omega_c} \left|H_{ja}(\omega)\right|^2 S_a(\omega)\, d\omega}{\int_0^{\infty} S_a(\omega)\, d\omega}\right]^{1/2} = \frac{\sigma_r}{\sigma_a} \qquad (1)$$

To simulate turbulence the Gaussian Von Karman model was used with an rms gust intensity level of one fps (References 2 & 3). The scale of turbulence used in the analysis is 500 feet. In order to compute the $\bar{A}$ of the pilot acceleration response a cutoff frequency of 30 Hz was used.

The characteristic frequency, N_o, is the radius of gyration of the response power-spectral density curve with respect to zero frequency (Equation 2).

$$N_o^2 = \frac{\left(\int_0^\infty (\omega/2\pi)^2 S_j(\omega)d\omega\right)}{\left(\int_0^\infty S_j(\omega)d\omega\right)} \tag{2}$$

NASTRAN computes σ_r and N_o by the solution of the airplane equations of motion. A post processor may be written in order to compute $\bar{A}$ and load exceedances N(y). Frequency of exceedance, N(y), is the number of exceedances of y per unit time or distance flown, where y is any response quantity (Equation 3).

$$N(y) = N_o \left[P_1 \exp\left(-\frac{y}{\bar{A}b_1}\right) + P_2 \exp\left(-\frac{y}{\bar{A}b_2}\right)\right] \tag{3}$$

The first fuselage bending mode, whose frequency is 9.5 Hz, is shown in Figure 4. Note that the pilot station coincides with the forward node point. Transfer functions and power spectral densities of the aircraft with and without the active controls are given in Figures 5 thru 12. Most of the response at the pilot station without the active controls is due to the short period and the first wing bending modes. When the RIMS and CAS are incorporated in the equations of motion, the response due to the short period mode is markedly lowered but some of the high frequency responses are amplified. The net result in the $\bar{A}$ of the pilot station acceleration is a 37% reduction due to the control system dynamics, while the N_o values are higher. If the visceral response of the pilot is only frequency dependent, the ride quality is significantly improved by the RIMS and CAS system used in the analysis. Most of the improvement in the ride quality is due to the RIMS interaction alone.

The restart capability of NASTRAN has been used to study the ride quality by varying the gains in the feedback loops. Similar restarts may be made to vary the scale of turbulence or the gust spectrum at a fraction of the cost of the parent analysis.

TRANSIENT RESPONSE ANALYSIS

Due to the poor ride quality of the modern fighter aircraft a great interest has been generated in time domain analysis in parallel with the frequency domain analysis in order to evaluate the handling quality. Transient analysis by a Fourier method is available in NASTRAN. The loads defined as a function of time are transformed into the frequency domain by Fourier transform methods.

In the present analysis a (1-Cosine) gust profile with a critical gradient of 12.5 chord is used. The maximum gust velocity in the profile is 50 ft/sec. The NASTRAN restart capability was used to tune the gust response by varying the critical gradient.

Using the Fourier method, the single gust profile is replaced by a series of pulses with a period of 20 seconds. The forcing function is zero for some time interval to allow for the decay of the responses. In order to evaluate the inverse transform equal frequency intervals, method 0, is used.

The time histories of the relative rotations of the flaperon and the stabilator are presented in Figures 13 and 14, respectively. The time history of the pilot station displacement is shown in Figure 15. Most of the pilot station response is due to the short period mode with a first wing bending mode contribution superimposed upon it, as shown by the magnified view in Figure 16. The responses are well decayed before the next gust pulse hits the aircraft.

CONCLUSIONS

This paper shows that NASTRAN is an extremely effective tool for aeroelastic analysis.

A subsonic gust response analysis has been performed in order to evaluate its usefulness in the flight vehicle system design. In an earlier evaluation, NASTRAN flutter analysis capability has been found to be very satisfactory as it provides several state-of-the-art methods and also saves considerable amount of man-hours by avoiding duplication of the structural model by the static and dynamics group(Reference 1). In the pre-NASTRAN era it has been a normal practice in many aircraft companies to model the vehicle structure separately in the stress, flutter and gust response groups. The aerodynamic model is also duplicated within the groups. A consistent and systematic method of incorporating control system dynamics in the various engineering disciplines is also lacking. With the advent of the control configured vehicle, a new engineering discipline of aeroservoelasticity is emerging in a dominant engineering role in the early design phases of the aircraft structure and integrated control system. NASTRAN aeroelastic package with its integrated stress, flutter, gust response, and control systems interaction is a very powerful structural analysis tool. It is well tailored for interactive graphics environments with data base management systems. With the emphasis on aeroelastic tailoring and structural optimization in the aircraft design, NASTRAN is amply ready to play a central role.

The gust response analysis performed in this paper on a light weight fighter has given a great insight in the active control system design. The use of RIMS and CAS systems reduce the $\bar{R}$ of the pilot station acceleration responses by 37%. The random gust response due to the short period mode is significantly lowered. However, the response due to the structural modes either remains unchanged or some of the higher frequency modes are attenuated. This means that the present RIMS and CAS feedback loop gains and filters have to be modified to suppress the high frequency structural responses. This is an important revelation, which would have gone unnoticed in a customary rigid airplane control system analysis. Use of the active control system aggravates the dynamic loads on the flaperon, stabilator, etc., structures.

The transient response analysis of a fighter aircraft is also performed by NASTRAN to satisfy a point of view, which wants to look at the time history of the pilot response in addition to the frequency related behavior studied by the random gust analysis. It is quite possible that the pilot's discomforts are related to the jerks he feels while flying over the discrete impulses of the gust rather than a visceral response based on the frequency contents of the gust induced excitation. More work has to be done in this area in which NASTRAN can be used as a major design tool.

In all aeroelastic analyses, the checkpoint and restart capability should be used in order to study the influence of changing the scale of turbulence, gust spectrum, structural damping, feedback loop gains and filters, etc., on the responses. Restart procedure is very cost effective and should be further improved.

Generation of aerodynamic influence coefficients and a procedure to weight the aerodynamic forces and moments on each of the panels should be provided in order to match the test data. Additional plot capability should be provided to plot all the aerodynamic elements, spline fitted modes and aerodynamic pressure distributions. For the design of large aircraft, spacewise variation of the gust spectrum should be incorporated and coherency, cross-correlation and cross-spectral density should be calculated as gust response analysis. Mode acceleration method of response computation should be provided in the rigid format. A method of computing shear, bending moment and torque at wing stations or fuselage stations, etc., and plotting them as response quantities is also needed.

REFERENCES

1. Singh, A. K., "Applications of NASTRAN in Aeroelastic Analyses at Northrop," Proceedings of the MSC/NASTRAN User's Conference, March 15-16, 1979; and Finite Element News, July 1979.

2. Rodden, W. P., Harder, R. L., and Bellinger, E. D., "Aeroelastic Addition to NASTRAN," NASA Contractor Report 3094, March 1979.

3. Anon., "MSC/NASTRAN Aeroelastic Supplement," 1979.

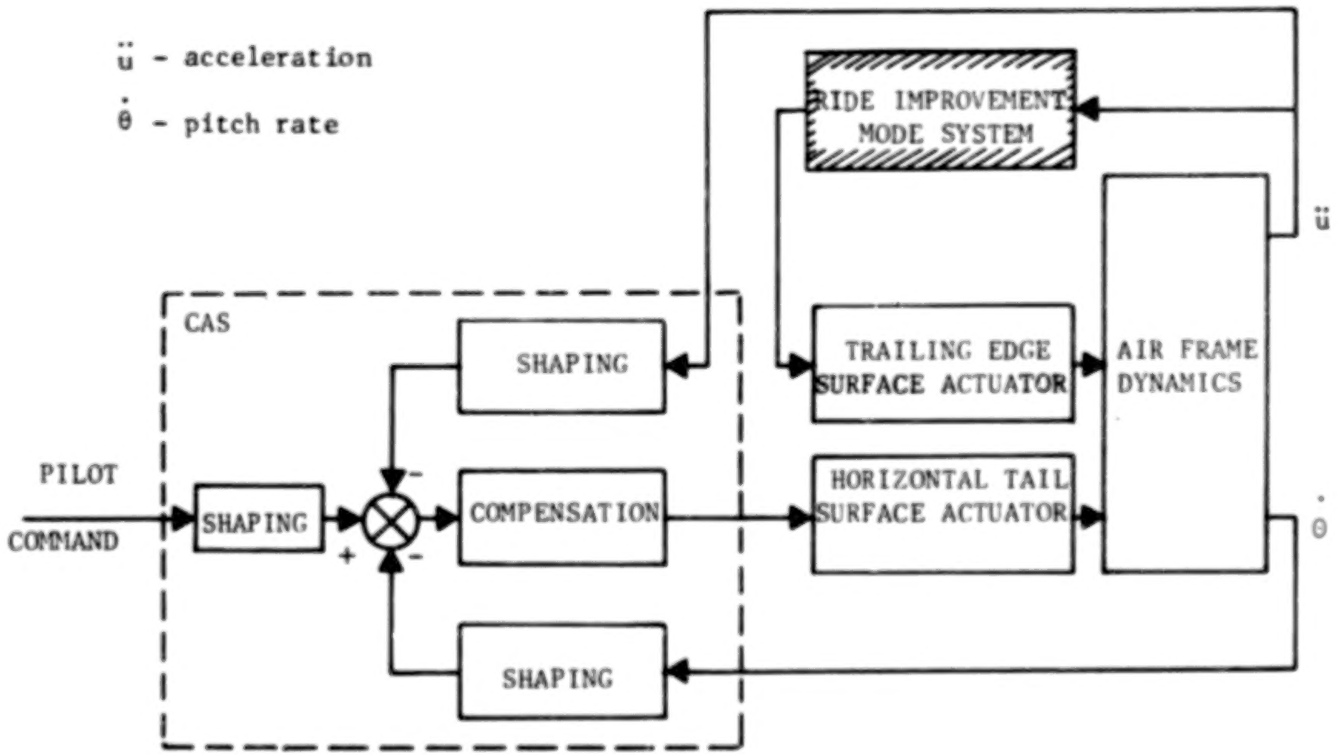

$\ddot{u}$ - acceleration
$\dot{\theta}$ - pitch rate
RIDE IMPROVEMENT MODE SYSTEM
$\ddot{u}$
CAS
SHAPING
TRAILING EDGE SURFACE ACTUATOR
AIR FRAME DYNAMICS
PILOT COMMAND
SHAPING
-
COMPENSATION
+
-
HORIZONTAL TAIL SURFACE ACTUATOR
$\dot{\theta}$
SHAPING

Figure 1. Longitudinal Control System Concept

BLANK PAGE

BLANK PAGE

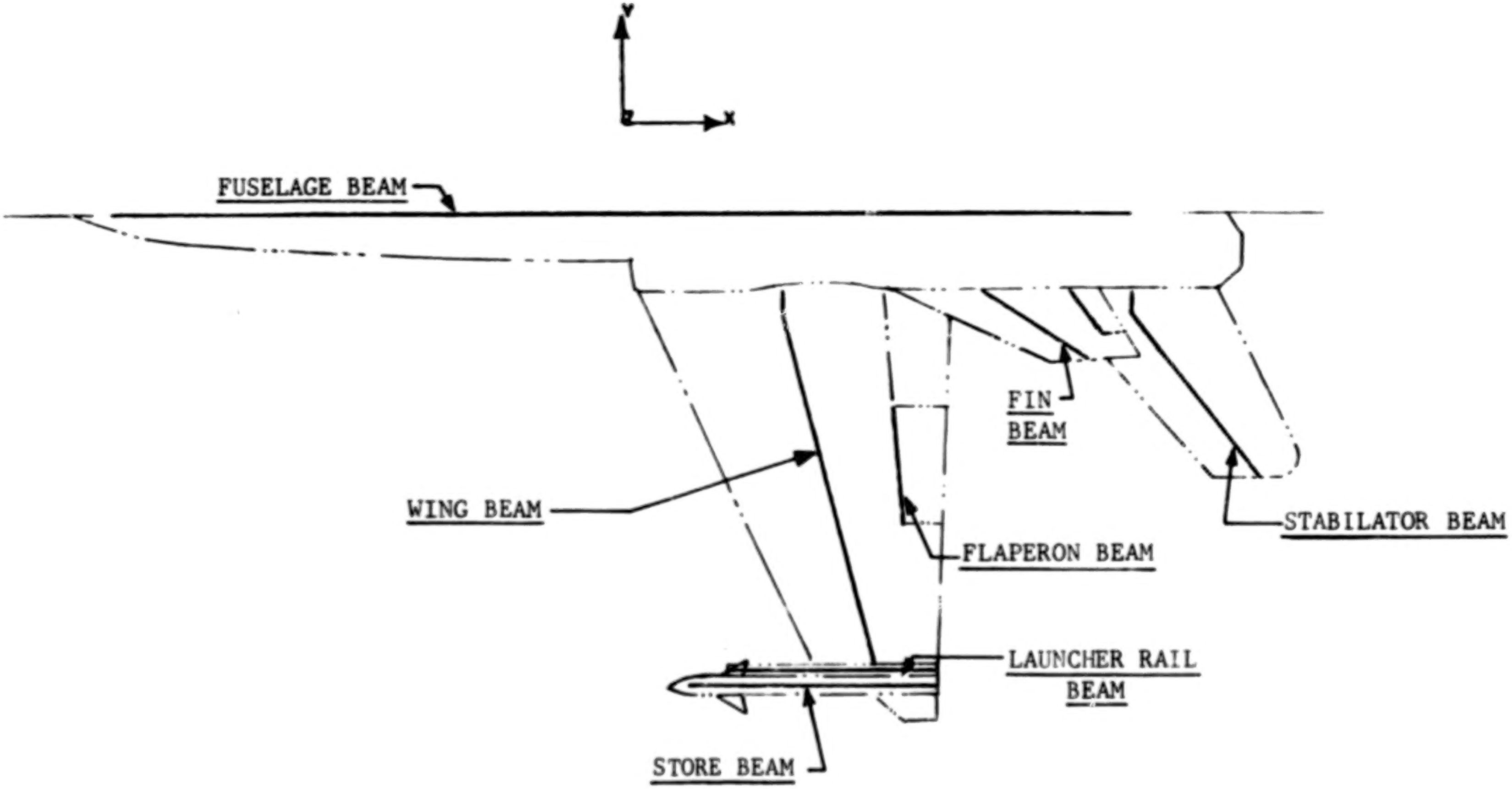

y
x
z
FUSELAGE BEAM
FIN BEAM
WING BEAM
STABILATOR BEAM
FLAPERON BEAM
LAUNCHER RAIL BEAM
STORE BEAM

Figure 2. Complete Airplane Structural Model - NASTRAN

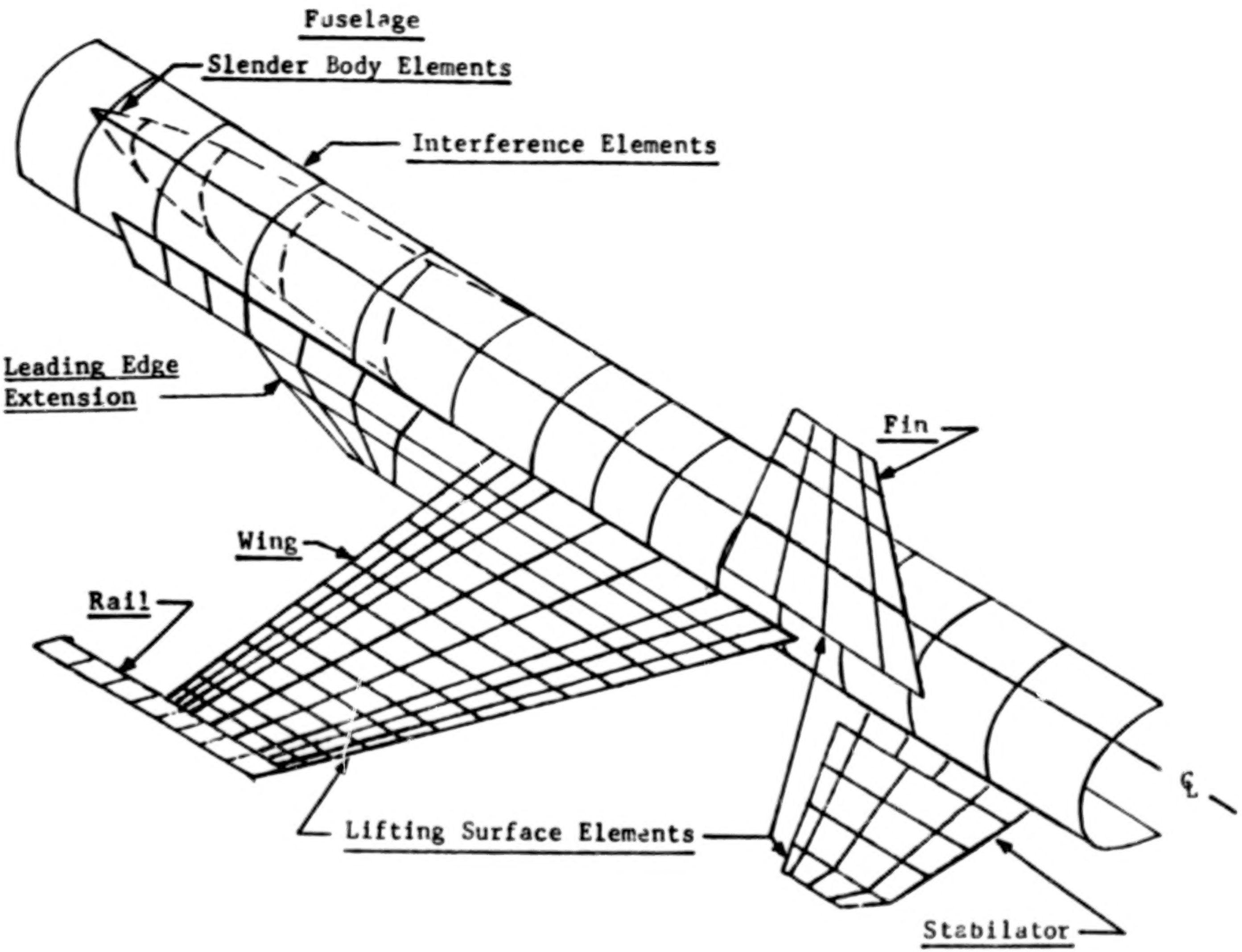

Fuselage
Slender Body Elements
Interference Elements
Leading Edge Extension
Fin
Wing
Rail
Lifting Surface Elements
Stabilator
℄

Figure 3. Complete Airplane Aerodynamic Model - NASTRAN

176

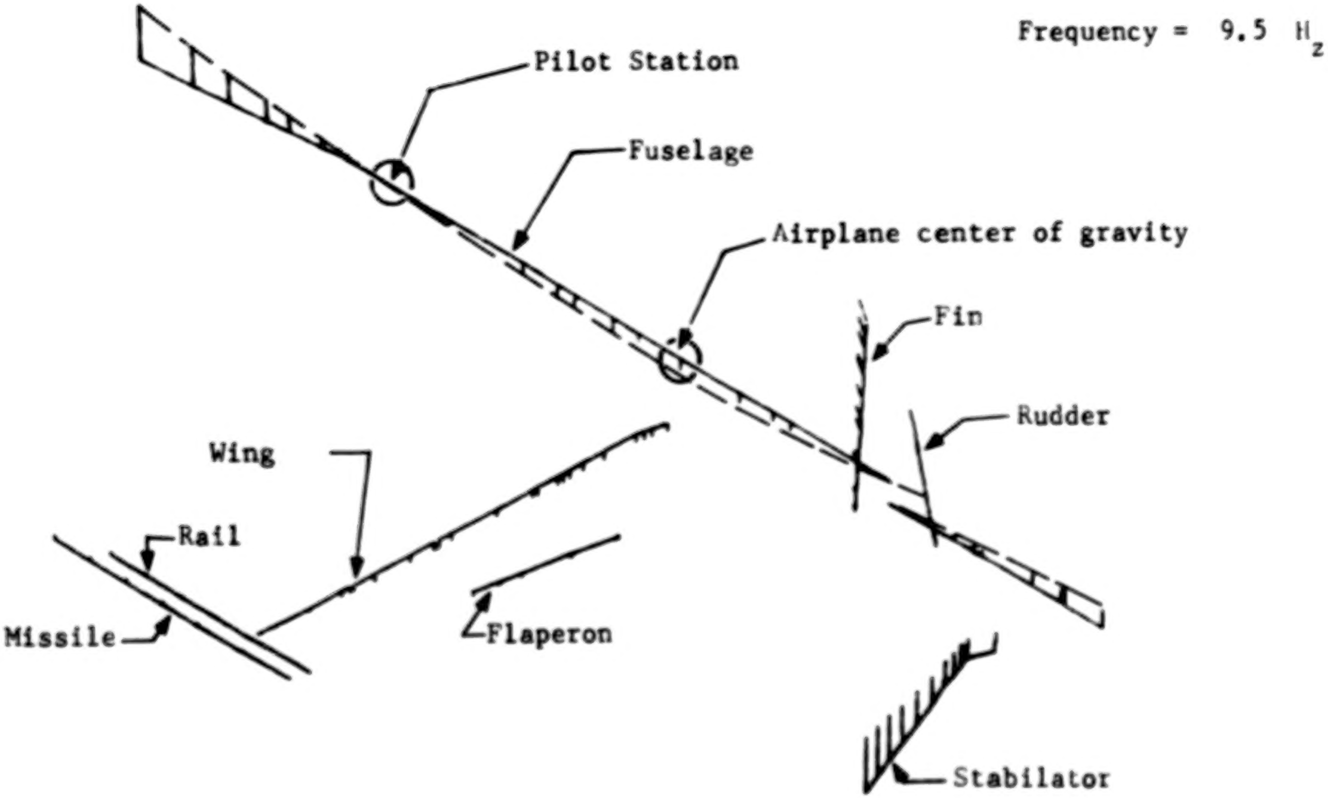

Frequency = 9.5 H_z
Pilot Station
Fuselage
Airplane center of gravity
Fin
Rudder
Wing
Rail
Missile
Flaperon
Stabilator

Figure 4. Fuselage Vibration Mode

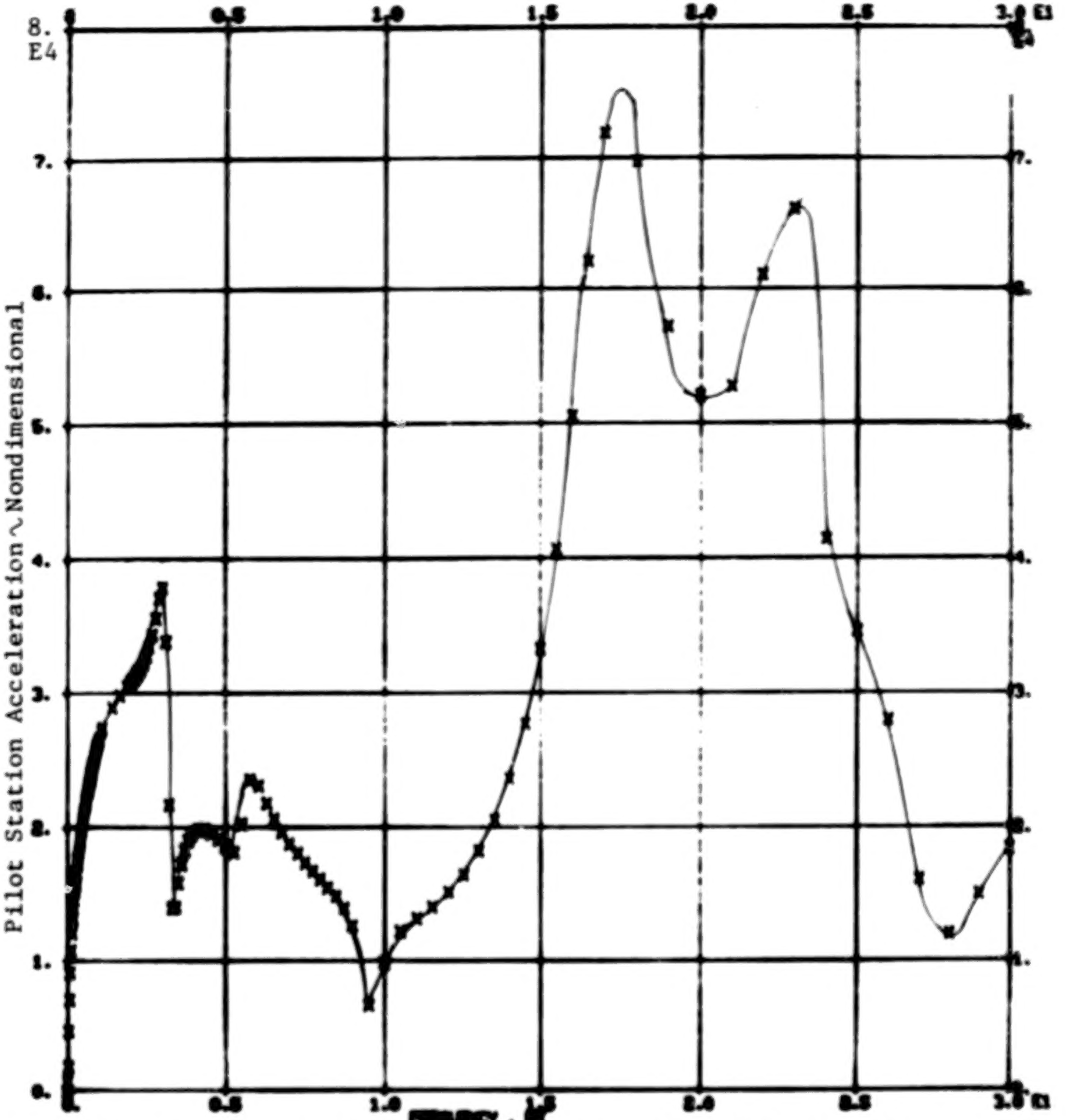


Figure 5. Frequency Response of Pilot Station Acceleration Without Active Controls

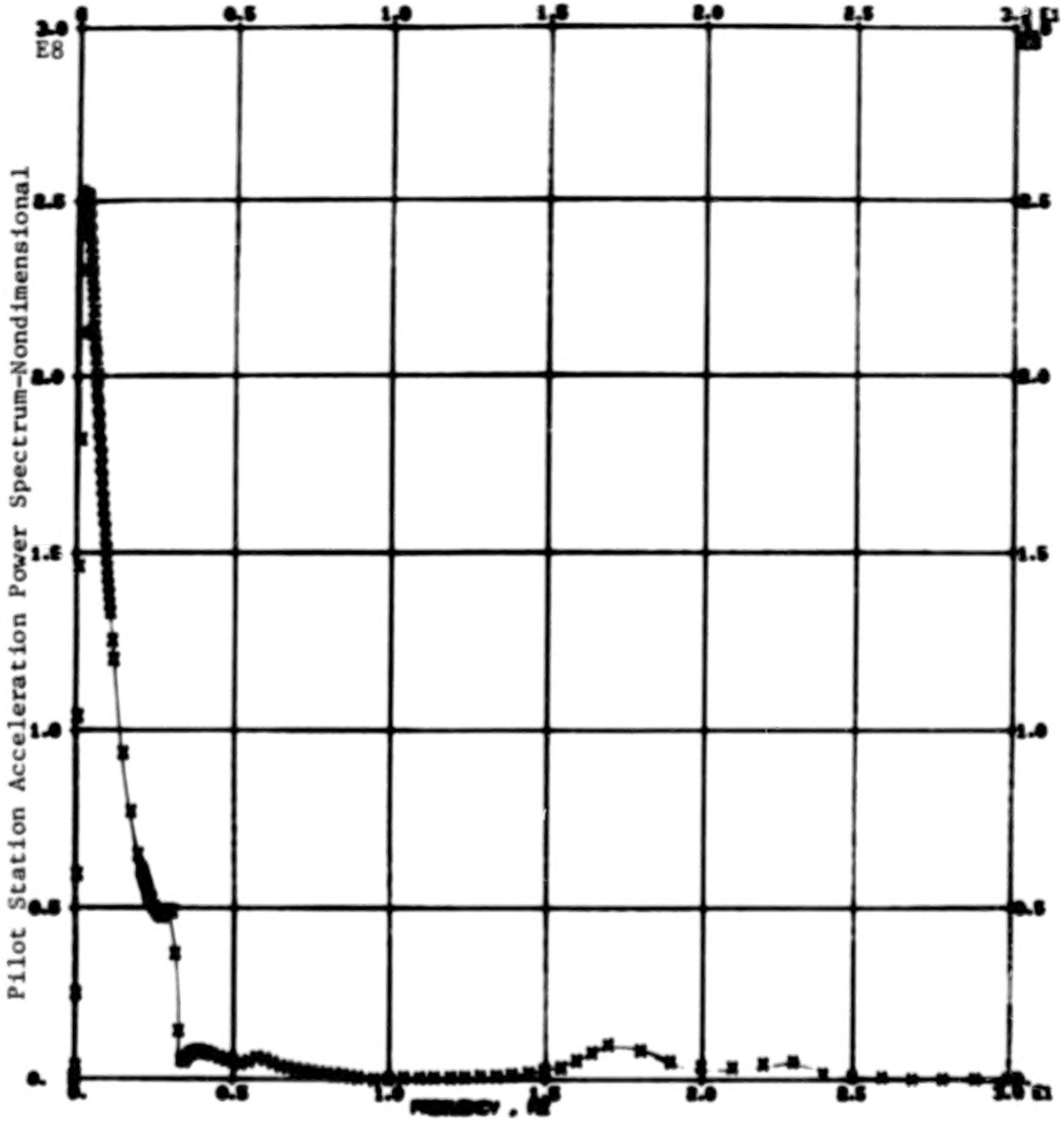


Figure 6. Power Spectral Density of Pilot Station Acceleration Without Active Controls

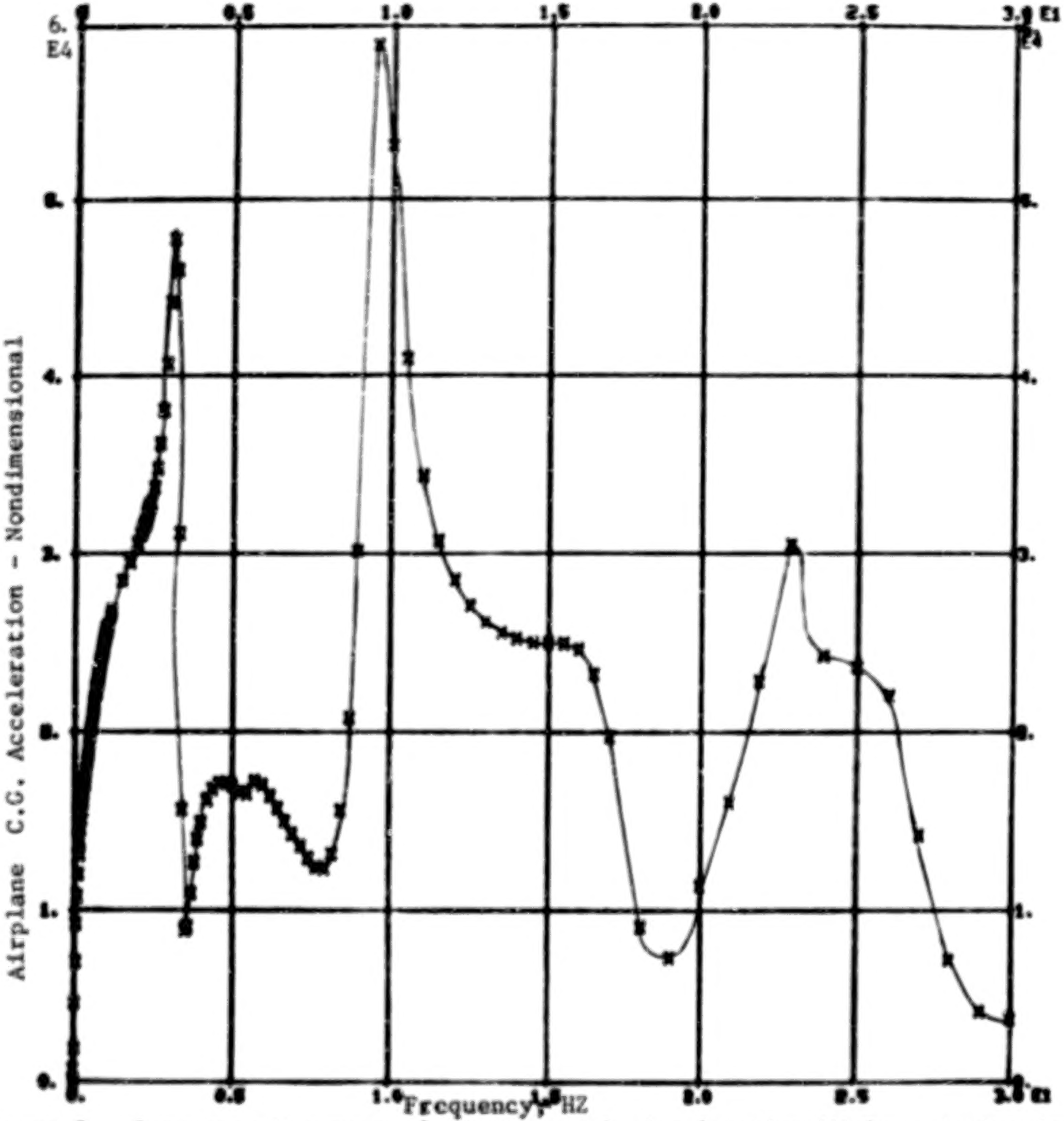


Figure 7. Frequency Response of Airplane C.G. Acceleration Without Active Controls

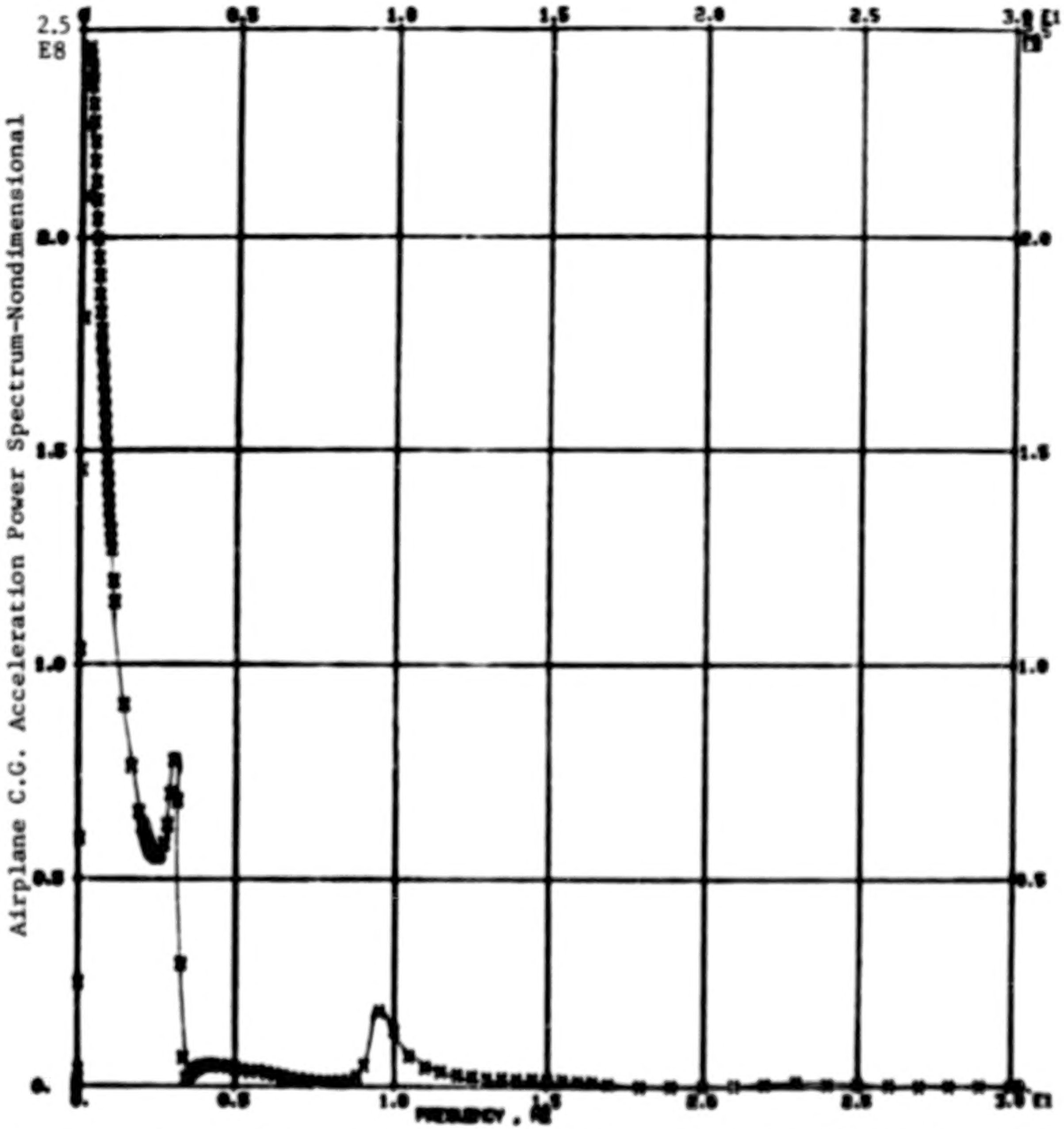


Figure 8. Power Spectral Density of Airplane C.G. Acceleration Without Active Control

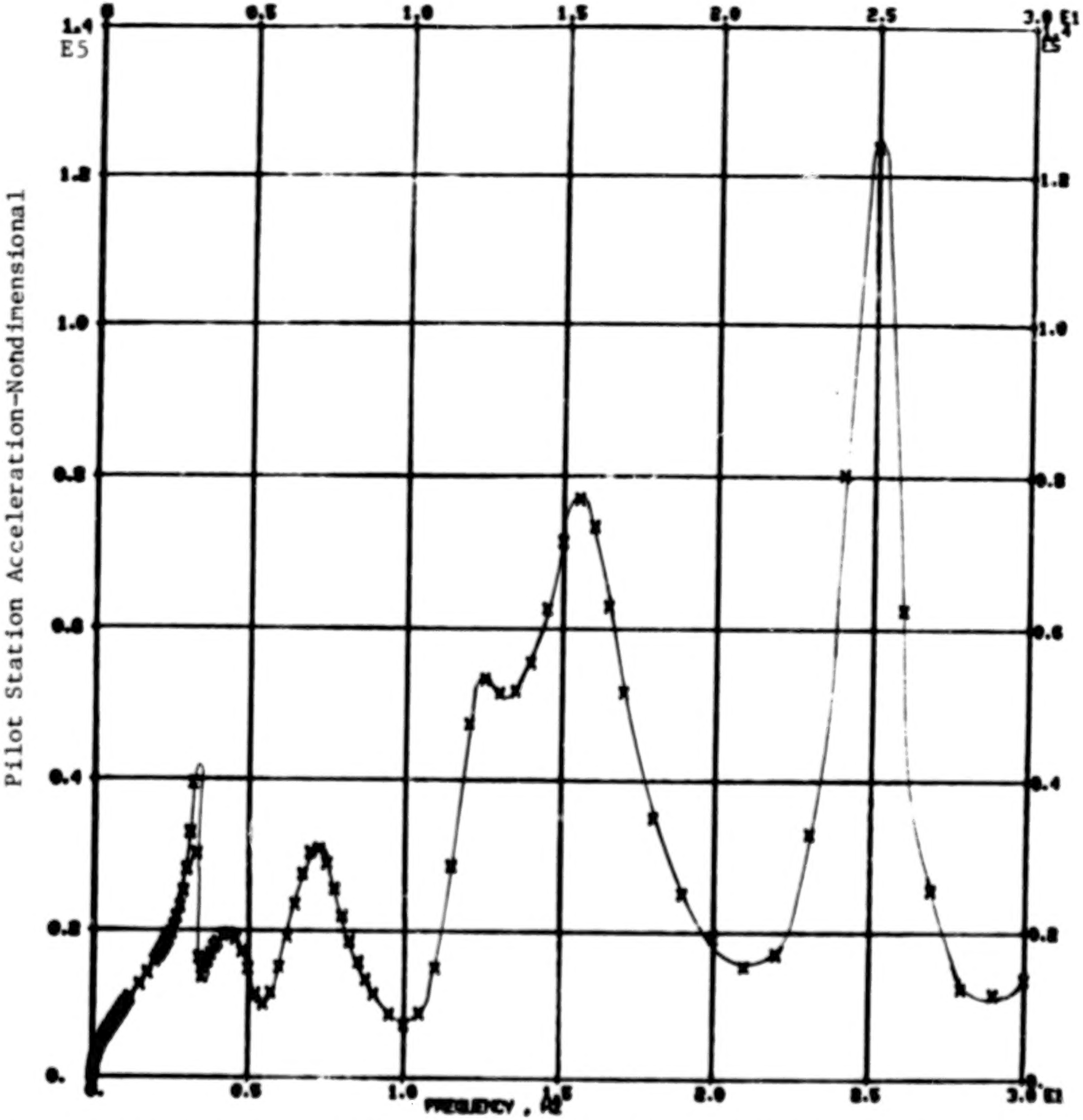


Figure 9. Frequency Response of Pilot Station Acceleration With Active Controls

182

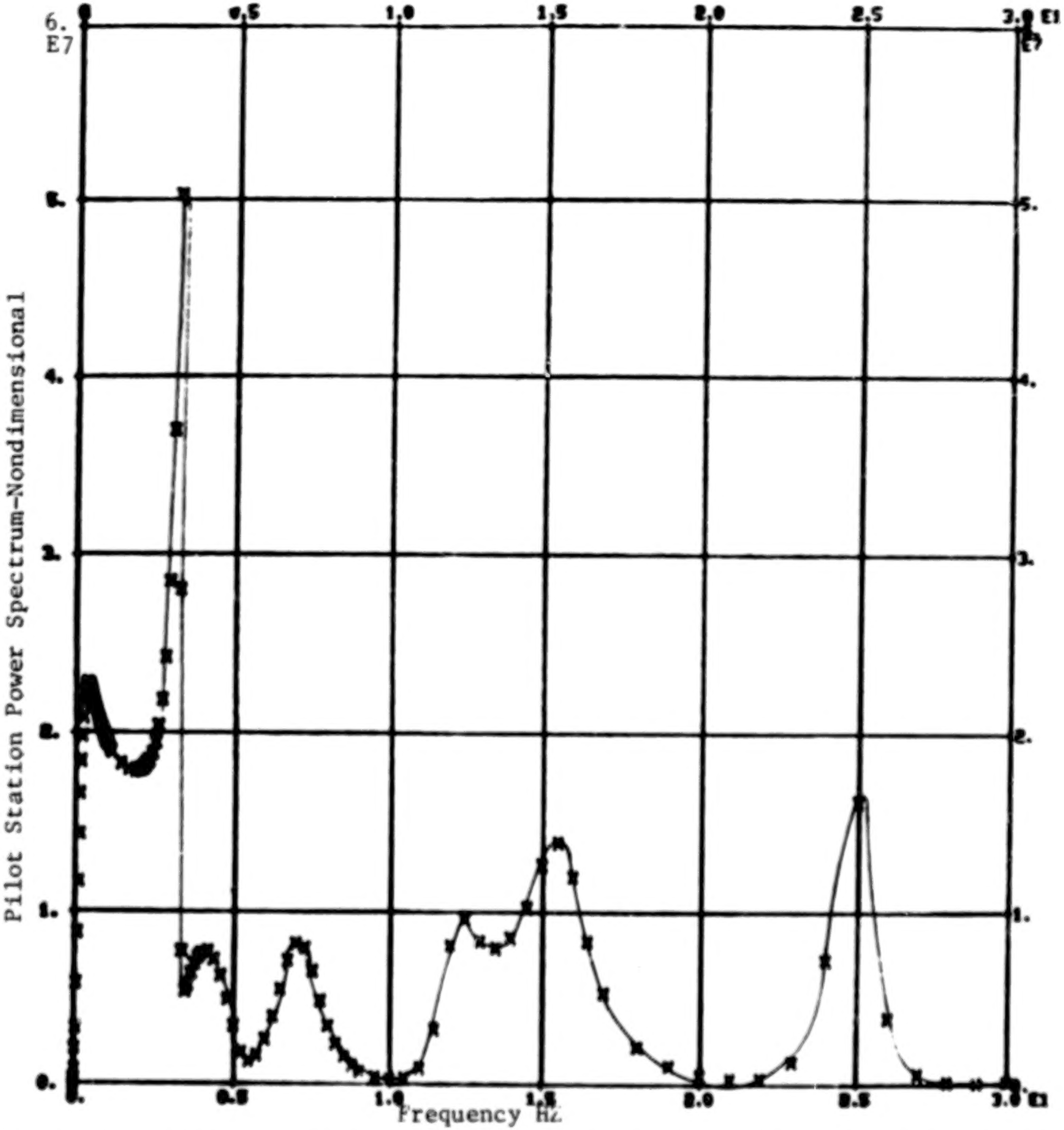


Figure 10. Power Spectral Density of Pilot Station Acceleration With Active Controls

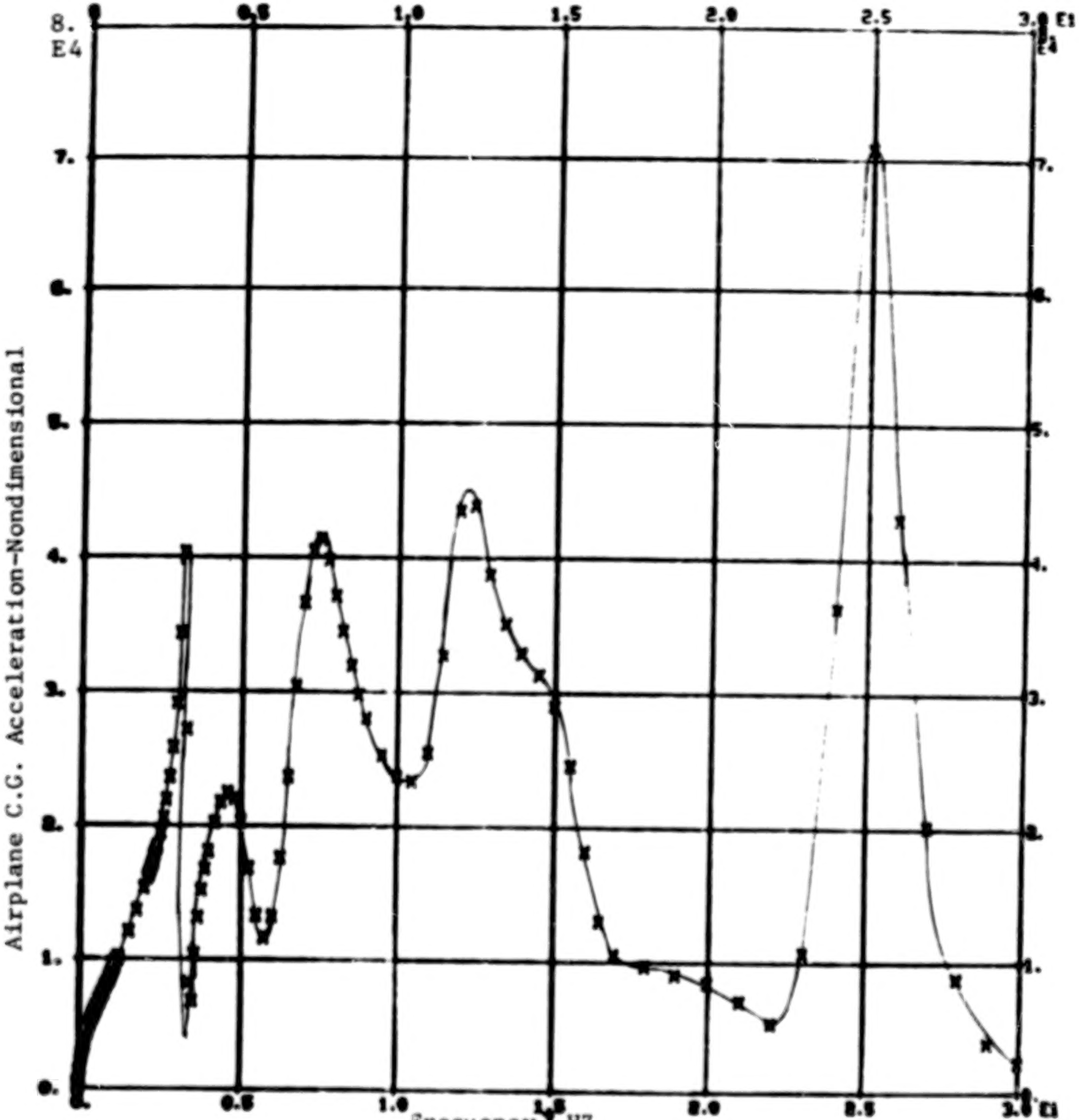


Figure 11. Frequency Response of Airplane C.G. Acceleration With Active Controls

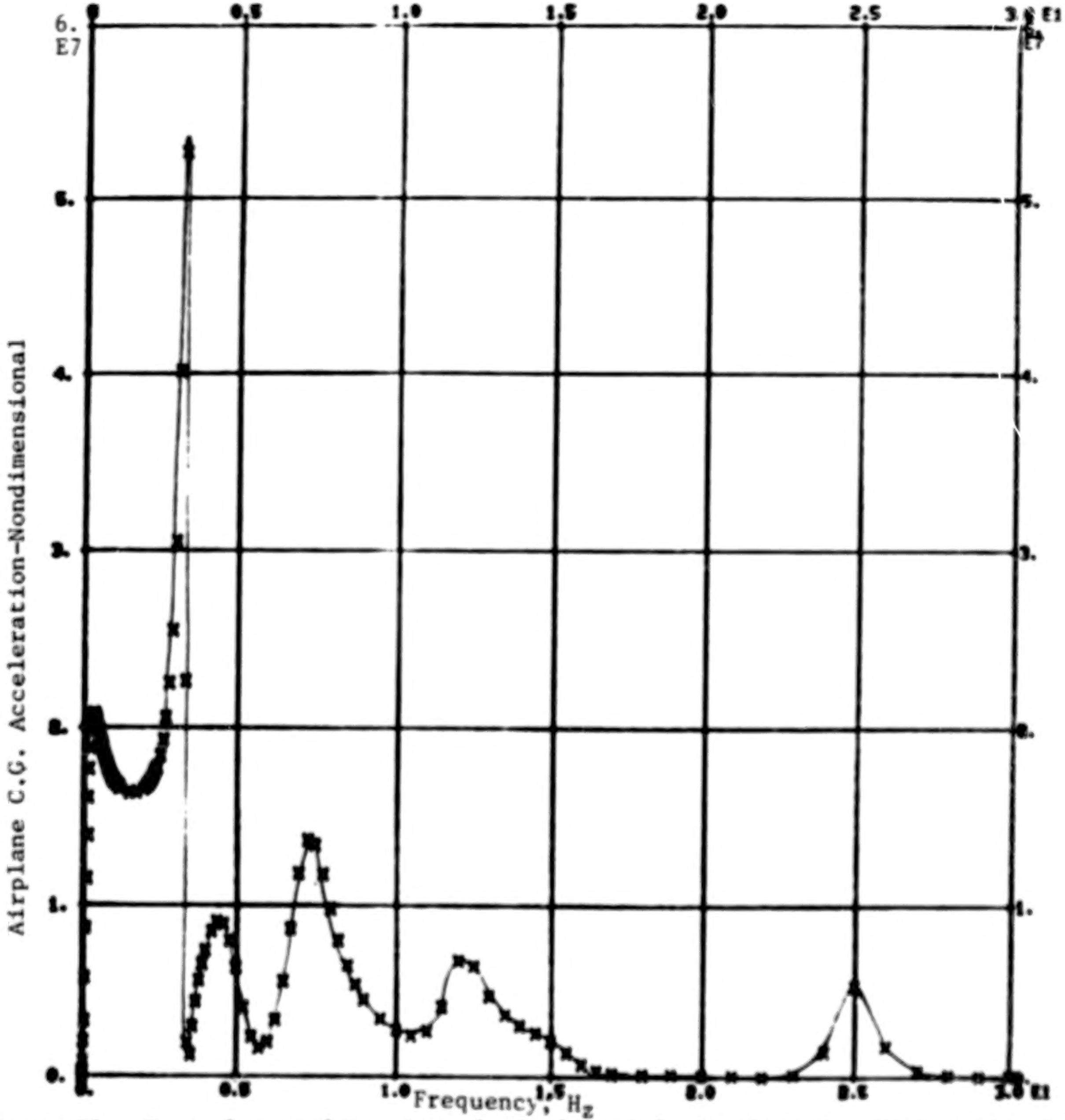


Figure 12. Power Spectral Density of Airplane C.G. Acceleration With Active Controls

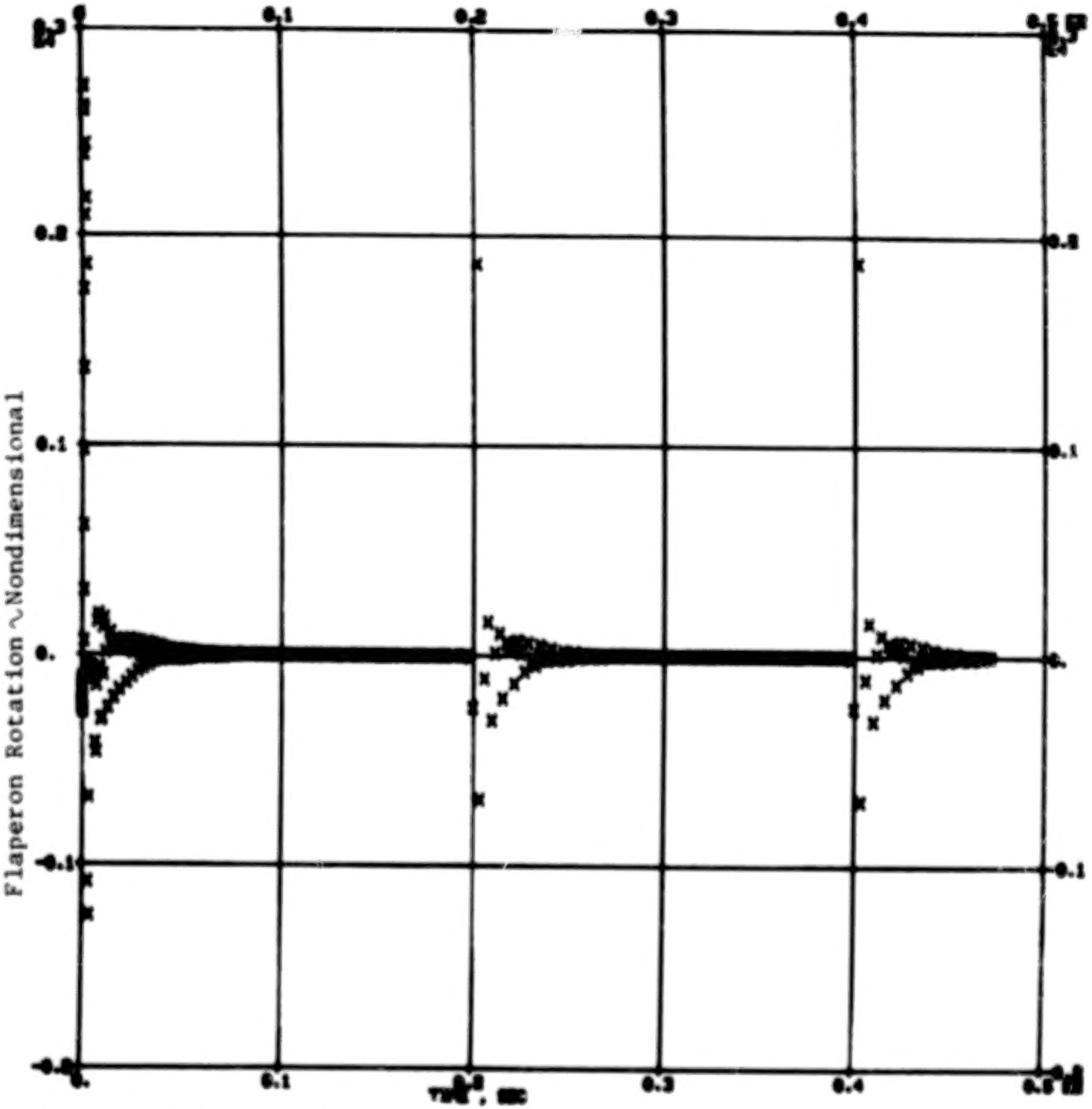


Figure 13. Time History of Flaperon Rotation With Active Controls

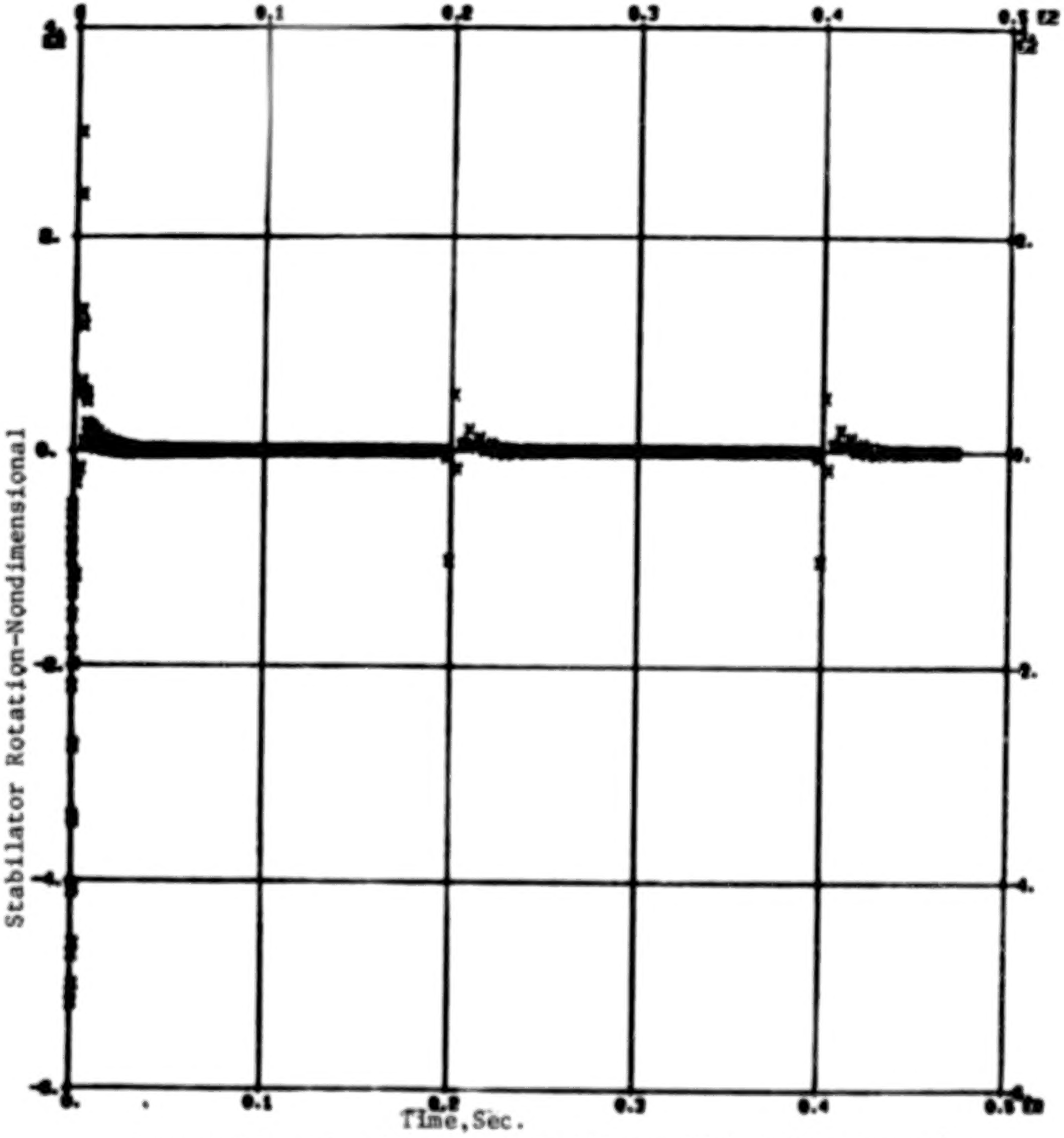
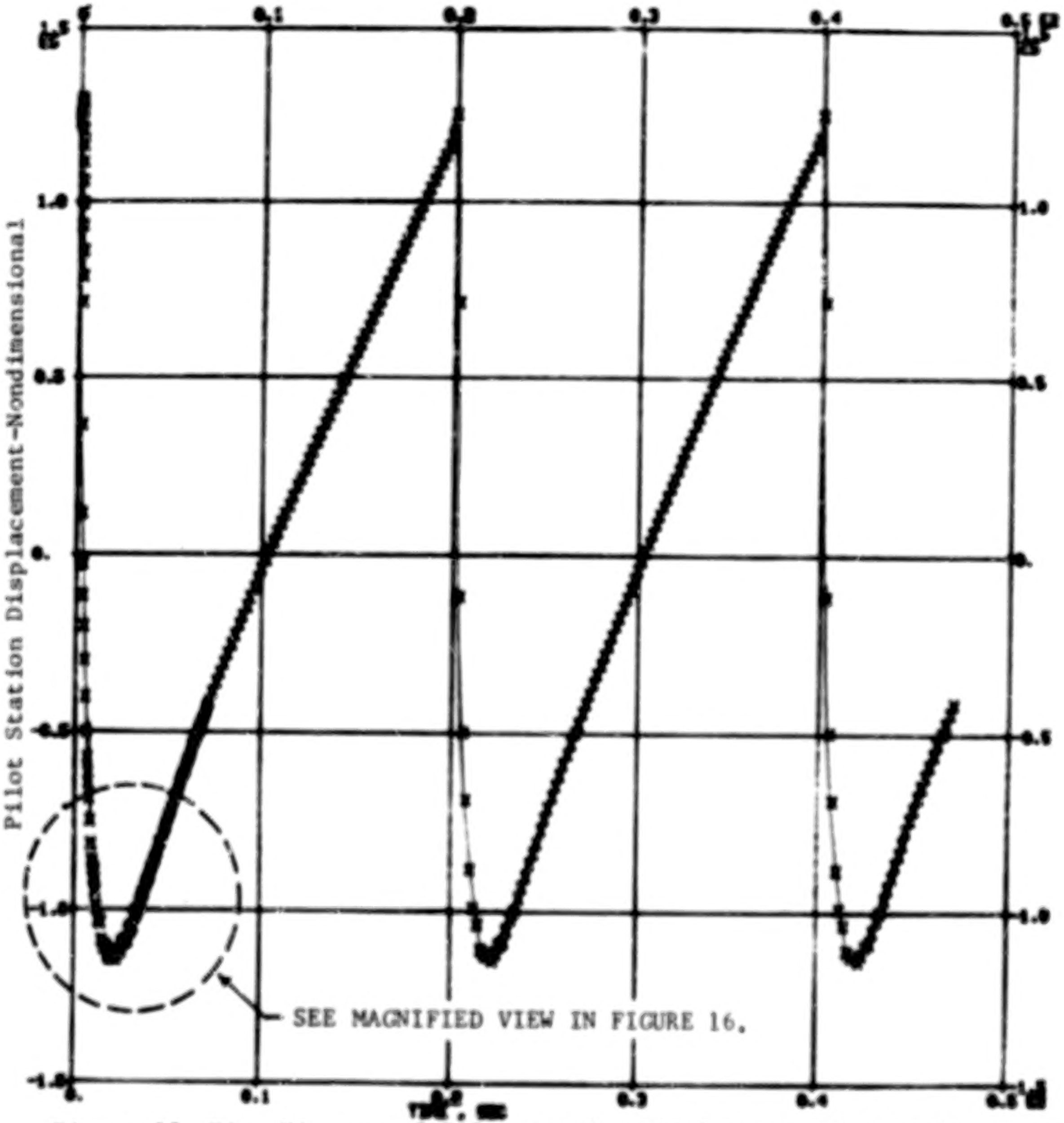


Figure 14. Time History of Stabilator With Active Controls



SEE MAGNIFIED VIEW IN FIGURE 16.

Figure 15. Time History of Pilot Station Displacement With Active Controls

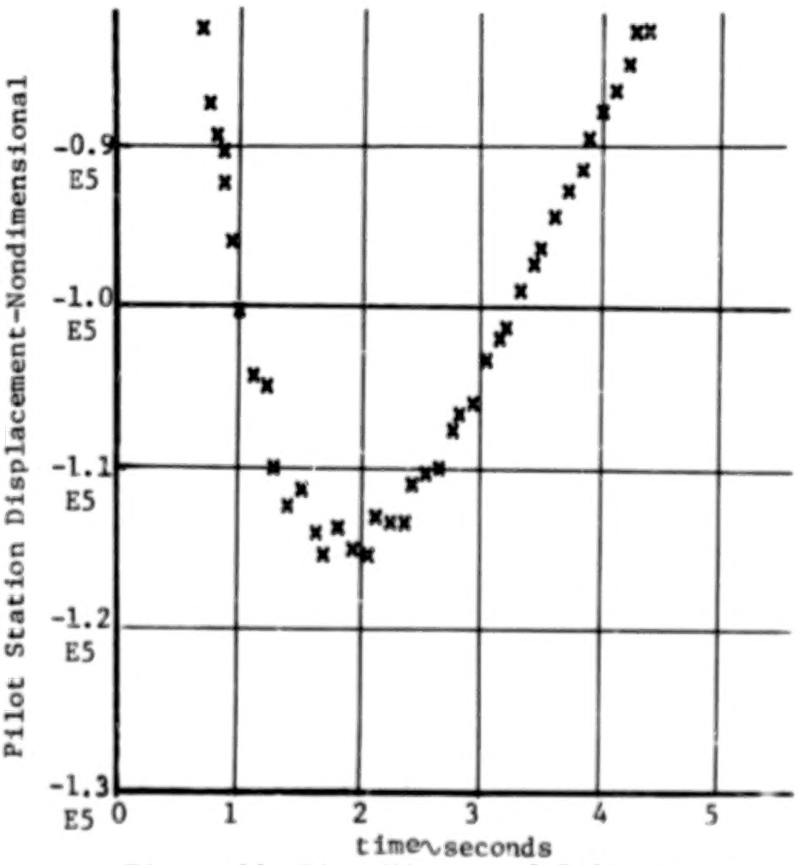


Figure 16. Time History of Pilot Station Displacement With Active Controls

IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS OPERATING SYSTEM*

Stephen S. Britten
M.I.T. Lincoln Laboratory

Betsy Schumacker
M.I.T. Department of Civil Engineering

SUMMARY

The NASA Structural Analysis (NASTRAN) computer program is operational on the IBM 360/370 series computers. While execution of NASTRAN has been described (ref. 1) and implemented under the virtual storage (VS) operating systems of the IBM 370 models, the IBM 370/168 computer can also operate in a time-sharing mode under the virtual machine (VM) operating system using the Conversational Monitor System (CMS) subset. This report describes the changes required to make NASTRAN operational under the CMS operating system.

"The views and conclusions contained in this document are those of the contractor and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States Government."

*This work is sponsored by the Department of the Air Force

INTRODUCTION AND BACKGROUND

M.I.T. Lincoln Laboratory first obtained NASTRAN in April 1974, when it purchased Level 15.5 from COSMIC for use on its newly installed IBM 370/168 computer. Minor modifications were made in order to make NASTRAN operational under the virtual storage concept of the IBM 370 system. However, much of the source code did not match the special executable load module code that had been received for direct loading onto IBM 3330 disk drives.

The Laboratory's NASTRAN capabilities were upgraded to Level 15.5.3+ in November 1975 with the assistance of NASA Goddard Spaceflight Center. The new source code and new executable load module code was in complete agreement. This Level 15.5.3+ of NASTRAN has adequately served the computational needs of the Laboratory since that original installation.

However, the Laboratory's computer configuration and operational schedule restrict the original version of NASTRAN to nighttime operation under the virtual storage (VS) batch-processing environment. The daytime operating mode (when engineering users are present) consists of an interactive time-sharing environment of up to 170 users under the virtual machine (VM) operating system using the Conversational Monitor System (CMS) subset.

It was desired to have NASTRAN available for general use in the interactive (CMS) as well as the batch (VS) environment, with commonality of input files. From an engineering user standpoint, the availability of NASTRAN in both environments is highly desirable. The CMS environment can be used for input data syntax checking, plotting of input and/or output data, and execution of relatively small analyses; while the VS environment can be used for execution of large analyses that require substantial computer time. The CMS environment is also useful to the programmer for maintenance and development of the many NASTRAN subroutines and functions.

For these reasons, the decision was made to develop a CMS/370 version of NASTRAN, working from the Laboratory's Level 15.5.3+ source. Compatability between the source files, the source listings, and the executable code under both the VS and CMS operating systems was required. Our primary objective was to make NASTRAN operational with minimal changes in computer coding.

This report describes the changes which were made to the NASTRAN computer program to develop a time-sharing version under 370-CMS and a compatible 370-VS1 batch-processing version. In this report, the following terminology is used:

1. CMS-NASTRAN: the version of NASTRAN developed at M.I.T. Lincoln Laboratory to operate under VM/370 (Virtual Machine facility for the IBM S/370) using the CMS (Conversational Monitor System) subset of VM/370.

2. VS1-NASTRAN: the version of NASTRAN developed at M.I.T. Lincoln Laboratory to operate under the VS1 (Virtual Storage) operating system for the IBM S/370.

The following definitions are based on the hardware and software developed by IBM for the implementation of the virtual storage concept.

Address Space - the set of memory addresses used by a program. VS1 allows a maximum total address space of 16 megabytes. CMS allows each user a possible maximum address space of up to 16 megabytes (limited at MIT-LL to 2 megabytes).

Page - a subdivision of the address space, 2K bytes on VS1 and 4K bytes on CMS.

Real Memory - the set of memory addresses which are physically available on the CPU. (3 megabytes on the Lincoln computer).

Virtual Memory - the address space which can be addressed by a relocating CPU. Physically, the virtual memory exists on a direct access storage device, and although a program may reference virtual memory in a random fashion, the information must be transmitted from virtual memory (direct access) to real memory one page at a time.

CMS DIFFERENCES AFFECTING NASTRAN

The CMS subset of the VM operating system is noticeably different from its VS operating system counterpart, in several areas:

(1) Input/output file structure,

(2) Load module formation, and

(3) Memory management and open core concepts.

These differences between CMS and VS required design decisions regarding implementation of NASTRAN under CMS. In all cases, we attempted to introduce minimal coding changes into the Level 15.5.3+ version of NASTRAN and tried to remain consistent with the original design philosophy regarding NASTRAN operation on IBM computers.

The following sections will deal in specifics concerning the abovementioned areas of difference and will describe our solutions to the problems encountered. The final section will mention some discovered coding errors as well as the implementation of a plotting package using the SC4060 plotter.

Input/Output File Structure

Execution of NASTRAN under the VS operating systems requires a large number of standard Job Control Language (JCL) cards. Modification of the Procedure (PROC) card that precedes these basic JCL cards allows the user flexibility to define space allocations, physical units, and program libraries to be searched.

The CMS operating system has NO Job Control Language cards. File definitions are performed by issuing FILEDEF commands immediately prior to execution time. The FILEDEF command defines the physical device type and the characteristics of the file. All CMS files are always stored and retrieved in 800 byte blocks. We have chosen 6400 byte blocks for the NASTRAN files - a size which will physically fit two blocks per track on an IBM 3330 disk drive as well as be wholly compatible with CMS block sizes. These FILEDEF commands are stored into an EXEC file which can be invoked by a user along with other commands to load and execute CMS-NASTRAN. (refer to Appendix A).

In order to minimize the coding changes required in GNFIAT (the program that generates the file allocation tables), it was necessary to provide a list of acceptable file names as input to the program. Under the VS operating system, acceptable names are chosen from the JCL cards included in the data deck. In CMS, the first card in the IOLIST file indicates (4I2 format) the number of acceptable (permanent, primary, secondary, and tertiary) file names, and the names themselves follow (refer to Appendix B).

Since file chaining is accomplished by CMS, the SPACE limitations needed for VS operations are no longer pertinent under CMS. For this reason, all IOLIST files were allowed to expand in additional 6400 byte blocks as often as needed to provide file space. The VS1-NASTRAN version uses the same file extension scheme outlined (ref. 2) for NASTRAN on the IBM 360-370 operating system.

The POINT macro differs in its operation under CMS from that under VS and thus coding changes were necessary for successful operation of BSAM I/O processing in CMS-NASTRAN.

Load Module Formation and Usage

The most significant difference concerning load module formation and usage under the CMS operating system is that no provision exists in CMS for an overlay structure. The NASTRAN load module generation under VS attempts to minimize the core requirements of each control routine LINKNSii, ii = 01, ..., 14 by use of overlay structures. Program core space in a virtual machine no longer becomes so critical, and thus provisions for overlay structure under CMS were felt not to be important.

As a simple first approach, the load module generation of all fourteen control routines LINKNSii was carried out with no overlays. While LINKNS01 required 660K for storage, the next largest, LINKNS13, required only 530K. NASTRAN, the super-link module, needs 45K and the system consumes 129K. Allowing for a reasonable work space in open core and providing FORTRAN with sufficient buffer space, CMS-NASTRAN thus requires a minimal core of 1200K. Our present account structure provides us with 2048K which we fully utilize while running CMS-NASTRAN. However, we are eyeing the possibilities of reducing core requirements to 1024K by splitting the larger LINKNSii control routines into multiple modules.

Load module formation under CMS also differs from the process of linkage editing used by VS. The linkage editor under CMS first searches the user's directory for TEXT files with the same names as CSECT entry names and then searches user-specified libraries called TXTLIBs in a specified order. Unlike partitioned-data sets (PDSs) on VS where member names are user-specified, the CMS main entry in a TXTLIB consists of the name of the first CSECT in the program, irrespective of the original name of the compiled program. For this reason, FORTRAN subroutines in NASTRAN containing only BLOCK DATA statements and named COMMONs have the CSECT name of the first named COMMON when compiled and thus that entry name when stored in TXTLIBs under CMS, but have the assigned member name when stored in OBJECT PDS's under VS.

In VS, the linkage editor has input cards - such as INCLUDE and INSERT - which specify specific members to be included in a module from a PDS and specify specific placement of a CSECT within a load module. It was this ability of the linkage editor to have the user specify placement within a module which enabled the developers of NASTRAN to implement the open core concept on IBM batch systems.

In CMS, there is no linkage editor, per se, but rather a loader with two user functions associated with it - LOAD and INCLUDE. The loader builds its own CSECT list which can be resolved from only two places: TEXT files or TXTLIBs. All TEXT file names and entry names in TXTLIBs must therefore be CSECT names. Since the loading process is primarily a search process, the order of CSECTs in a module will depend on the order in which programs were encountered in the search. This absence of user-specified ordering in CMS loading required the removal of certain CSECTs (CONMSG, MAPFNS, EJDUM2, and some named COMMONs) from TXTLIBs and their storage in TEXT files with file names different from CSECT names. These CSECTs will be unresolved externs after the LOAD function but will then be put at the end of the module in a user-specified order by means of the INCLUDE function, making sure EJDUM2 is last since it represents open core.

The CMS linkage editor also differs from its VS counterpart in that it resolves COMMON CSECTs at execution time rather than at load time. For this reason, a way had to be found to force those FORTRAN subroutines containing only BLOCK DATA statements and named COMMONs to be loaded with the other CSECTs.

This was accomplished by writing additional FORTRAN subroutines LINKiiC that did nothing more than CALL the named COMMONs whose CSECT entries existed in the TXTLIBs. This forced inclusion at load time. (Refer to Appendix C.)

A change was also required to LINKNSii and NASTRAN to reference the proper entries in the current FORTRAN function library (e.g. IHOERRM, etc.).

The lengths of all load modules formed in the CMS system are stored in a MASTER file. These lengths can than be used to control the open core requirements described in the next section. Load module generation under CMS reserves the first 128K (origin hex 20000) for the operating system and CMS. All load modules have the user's lowest address origined to 20000, which for NASTRAN execution meant loading the super-link NASTRAN at location 20000. In order to allow for future growth and/or modifications to the super-link NASTRAN, the origin for all LINKNSii entries was taken to be hex location 2AD00. This LINK origin is stored in the MASTER file along with the total load module lengths of all fourteen LINKNSii modules (refer to Appendix D).

Memory Management and Open Core Concepts

Memory management and open core concepts under the VS operating system have been very clearly presented (ref. 1) and will not be reviewed here. Instead, the manner in which the CMS operating system must handle these tasks will be explained.

Under CMS, the super-link NASTRAN load module is initially loaded into core at origin hex 20000 and execution begins. (Figure 1 depicts address space allocation for CMS-NASTRAN execution.)

1. Within the NASTRAN module, a user request for working storage (GETMAIN) will be issued for all of available memory.

2. The NASTRAN module then releases FOURK (16K words under CMS) high address space for operating system use via the FREEMAIN macro.

3. All of the GETMAINed area will be managed by NASTRAN rather than by CMS for all executions of LINKNSii load modules requested by the super-link program NASTRAN.

4. The load modules LINKNS01 thru LINKNS14 have been generated at origin 2AD00 and are loaded with the CMS macro LOADMOD instruction as they are needed.

The above memory management scheme was required since the CMS simulation of VS LINK, GETMAIN, and FREEMAIN macros did not present a duplicate image of core to NASTRAN of that which VS presented. Thus, the macro for loading a module into core could not be LINK but had to be LOADMOD. This in turn said memory had to be obtained prior to and for the LOADMOD. Since the LOADMOD forced CMS to obtain buffer space for itself, we were forced to resort to a single GETMAIN with total core management performed by NASTRAN so that fragmentation of memory (into NASTRAN blocks intersperced with system blocks) was eliminated.

Since CMS-NASTRAN has now assumed the responsibility of managing memory, the "open core" concept of data management must also be assumed by NASTRAN. Under the overlay load module structure of VS, "open core" at the end of an overlay tree segment was denoted by a dummy named-COMMON section. This named-COMMON could contain an array whose length extended into the open core region but yet would not destroy or overwrite code or data in other segments of the overlay. Unfortunately, under CMS with no overlay structure, we are faced with numerous dummy named-COMMONs, each of which must be located near the end of the load module so that it will not overwrite other code and/or data in the module. The placement of these named-COMMONs is accomplished by INCLUDing newly-written FORTRAN subroutines, LINKiiCC, when doing the load module generation (refer to Appexdix E).

Minor Differences

Several subroutines contain DATA statements which are used to initialize variables which are subsequently modified. The DATA statements were used assuming that the programs containing these statements exist on segments of the overlay tree in VS, implying that a fresh copy of the subroutine will be loaded (and thus reinitialization of DATA variables will occur) each time the segment is needed. Under CMS, the subroutine when it is reentered contains the last values of the DATA variables and they are not reinitialized, thus causing errors. This problem would also occur in a non-overlay VS version of NASTRAN.

The graphics package acceptable to the CMS environment at M.I.T. Lincoln Laboratory utilizes SC4060 meta code to generate the meta code for on-line Tektronix terminals. For this reason, a SC4060 plotting package was added to the NASTRAN package. Since the CMS environment provides disk file facilities for graphics as well as standard I/O, restrictions on graphics files in NASTRAN were removed so that disk file definitions for PLT1 and/or PLT2 files are permissible.

Conclusions

The implementation of CMS-NASTRAN has been completed. While operation has been limited to test examples and several small production runs, it definitely shows promise as a useful program on future projects in which

NASTRAN will be needed. Program enhancements (such as the SC4060 graphics package) are easily made while working in the CMS environment and can be added to the VS-NASTRAN after successful debugging has been completed.

A new VS1-NASTRAN is being generated at the present time. It will embody the design philosophy of CMS-NASTRAN and, thus, will not be in overlay structure. Compatibility of source (except for several assembly language programs) and executable code between the two systems should provide for the reliability needed to assure identical results in either mode.

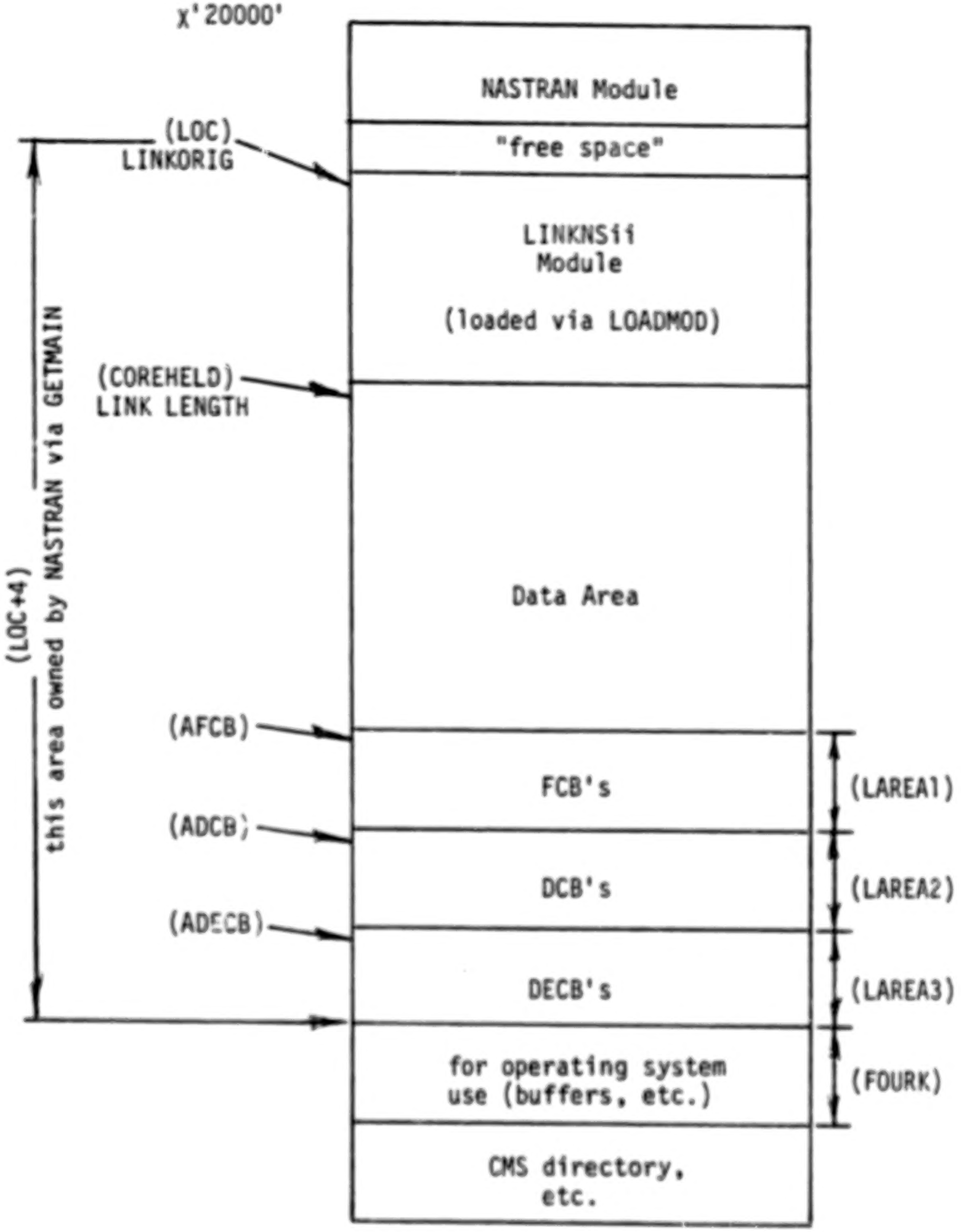
It is anticipated that CHKPNT saves and restarts should be possible between the CMS and VS operating systems. This would allow preprocessing (data syntax checking, undeformed structural plots, etc.) to be performed and check pointed under CMS; restored, executed, and checkpointed under VS; and finally restored and postprocessed (including deformed structural plots) under CMS.

REFERENCES

1. McCormick, C. W. and Redner, K. K.: "Study of the Modifications Needed for Effective Operation of NASTRAN on IBM Virtual Storage Computers", NASA CR-2527, 1975.

2. Anon. The NASTRAN Programmer's Manual (Level 15.5), NASA SP-223(01), May 1973.

Figure 1 CMS Allocation of Address Space For NASTRAN


X'20000'
NASTRAN Module
"free space"
(LOC)
LINKORIG
LINKNSii
Module
(loaded via LOADMOD)
(COREHELD)
LINK LENGTH
(LOC+4)
this area owned by NASTRAN via GETMAIN
Data Area
(AFCB)
FCB's
(LAREA1)
(ADCB)
DCB's
(LAREA2)
(ADECB)
DECB's
(LAREA3)
for operating system
use (buffers, etc.)
(FOURK)
CMS directory,
etc.

Appendix A: Partial Listing of NASTRAN EXEC File

```
&CONTROL OFF NOMSG
-START &TYPE ENTER FILENAME OF NASTRAN FILE
&READ ARGS
&IF &INDEX NE 1 &GOTO -START
&FN = &1
-CONT &TYPE ENTER FILEMODE FOR NASTRAN OUTPUT FILES
&READ ARGS
&IF &INDEX NE 1 &GOTO -CONT
&FM = &1
FILEDEF 1       DISK     FT01  NAST     &FM     (BLOCK   80 RECFM F LRECL   80
FILEDEF 4       DISK     FT04  NAST     &FM     (BLOCK  133 RECFM F LRECL  133
FILEDEF 5       DISK     &FN   NASTRAN
FILEDEF 6       DISK     &FN   OUTPUT   &FM     (BLOCK  133 RECFM F LRECL  133
FILEDEF 7       DISK     &FN   PUNCH    &FM     (BLOCK   80 RECFM F LRECL   80
FILEDEF POOL    DISK     POOL  NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF NPTP    DISK     NPTP  NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF OPTP    DISK     OPTP  NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF PLT2    DISK     &FN   4060     &FM     (BLOCK  800 RECFM F LRECL  800
FILEDEF PRI01   DISK     PRI01 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF PRI02   DISK     PRI02 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF PRI03   DISK     PRI03 NAST     &FM     (BLOCK 6400 RECFM F
           •                                              •
           •                                              •
           •                                              •
           •                                              •
FILEDEF PRI30   DISK     PRI30 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF PRI31   DISK     PRI31 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF PRI32   DISK     PRI32 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF SEC01   DISK     SEC01 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF SEC02   DISK     SEC02 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF SEC03   DISK     SEC03 NAST     &FM     (BLOCK 6400 RECFM F
FILEDEF TER01   DISK     TER01 NAST     &FM     (BLOCK 6400 RECFM F
LOADMOD NASTRAN
START
```

Appendix B: Listing of NASTRAN IOLIST File

```
04320301
PERMPOOL
PERMNPTP
PERMOPTP
PERMPLT2
PRIMPRI01
PRIMPRI02
PRIMPRI03
PRIMPRI04
PRIMPRI05
PRIMPRI06
PRIMPRI07
PRIMPRI08
PRIMPRI09
PRIMPRI10
PRIMPRI11
PRIMPRI12
PRIMPRI13
PRIMPRI14
PRIMPRI15
PRIMPRI16
PRIMPRI17
PRIMPRI18
PRIMPRI19
PRIMPRI20
PRIMPRI21
PRIMPRI22
PRIMPRI23
PRIMPRI24
PRIMPRI25
PRIMPRI26
PRIMPRI27
PRIMPRI28
PRIMPRI29
PRIMPRI30
PRIMPRI31
PRIMPRI32
SECOSEC01
SECOSEC02
SECOSEC03
TERTTER01
```

Appendix C: Listing of Typical LNKiiC FORTRAN Subroutine

```
      SUBROUTINE LNK01C
C
C     SUBROUTINE RPDABD
          CALL XSFA1
C     SUBROUTINE IFXDBD
          CALL IFPDTA
C     SUBROUTINE IFX0BD
          CALL IFPX0
C     SUBROUTINE IFX1BD
          CALL IFPX1
C     SUBROUTINE IFX2BD
          CALL IFPX2
C     SUBROUTINE IFX3BD
          CALL IFPX3
C     SUBROUTINE IFX4BD
          CALL IFPX4
C     SUBROUTINE IFX5BD
          CALL IFPX5
C     SUBROUTINE IFX6BD
          CALL IFPX6
C     SUBROUTINE IFX7BD
          CALL IFPX7
C     SUBROUTINE IFPABD
          CALL IFP1A
C     SUBROUTINE AXICBD
          CALL IFP3BD
C     SUBROUTINE XGPIBD
          CALL XGPIC
C     SUBROUTINE XMPLBD
          CALL XGPI2
C     SUBROUTINE XSORBD
          CALL XSRTBD
C     SUBROUTINE UMFZBD
          CALL UMFZZZ
C     SUBROUTINE XBSBD
          CALL XLKSPC
C
      RETURN
      END
```

Appendix D: Listing of NASTRAN MASTER File

```
LINKORIG 02AD00
LINKNS01 0D00A8
LINKNS02 08C580
LINKNS03 092F50
LINKNS04 072030
LINKNS05 094F00
LINKNS06 078218
LINKNS07 078428
LINKNS08 0976B0
LINKNS09 054950
LINKNS10 073AD8
LINKNS11 08D8D0
LINKNS12 05C688
LINKNS13 0AF908
LINKNS14 07A390
```

Appendix E: Listing of Typical LNKiiCC FORTRAN Subroutine

```
C     DUMMY TO FORCE XGPI1 TO COME JUST BEFORE EJDUM2 (OPEN CORE)
      BLOCK DATA
      COMMON DUM4(100)
      COMMON /GINOX/ DUM6(163)
      COMMON /SETUP/ DUM2(7)
      COMMON /IFP3LV/ DUM10(104)
C     BEGINNING OF DUMMY COMMONS THAT MARK START OF OPEN CORE.
      COMMON /IFP1X/ DUM9(371)
      COMMON /ENDSSS/ DUM7(2)
      COMMON /IFPXX/ DUM8
      COMMON /IFP3ZZ/ DUM11
      COMMON /IFP4ZZ/ DUM12
      COMMON /IFP5ZZ/ DUM13
      COMMON /XCSABF/ DUM14
      COMMON /ESOPT/ DUM
      COMMON /UMFXXX/ DUM1
      COMMON /ESFA/ DUM5
      COMMON /XGPI1/ DUM20(5)
      COMMON /EJDUM2/ DUM21
      END
```

BLANK PAGE

IFEMS, AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM

Spencer McKellip, Todd Schuman and Spencer Lauer
Sikorsky Aircraft
Division of United Technologies

SUMMARY

General purpose 3D finite element mesh generators, require detailed geometric description of the component to be modeled. For complex shapes, this process can be very time consuming. On the other hand, most general purpose CAD/CAM systems have the capability of defining geometry in an efficient, interactive environment. This paper describes one method of coupling a CAD/CAM system with a general purpose finite element mesh generator.

The problem is one of generality in application. Most interactive mesh generators are either tailor made to specific applications or are restricted to classes of geometry such as bodies of revolution, dragged shapes, 2D geometry or digitized input. What is needed is a completely general purpose, 3D interactive tool that can handle any conceivable geometry without digitizing or any other restriction.

A three dimensional interactive graphic system for defining geometry can be an expensive proposition for just finite element analysis. However such capabilities have been in existence for some time in the form of computer aided design and manufacturing systems, commonly known as CAD/CAM systems. These systems are logical candidates for front ends to a general purpose, 3D finite element mesh generator.

The system described in this paper consists of three programs:

1) CAIDS, A Sikorsky developed CAD system.

2) IFEMS, A Sikorsky developed interactive processor to couple a CAD system with a mesh generator.

3) PWAMESH, A Pratt & Whitney developed mesh generator.

INTRODUCTION

Since the introduction of finite element analysis in the 1960's and thru the 1970's it has become obvious that there was a clear need for powerful mesh generation techniques.

What evolved were two different approaches to mesh generation. One was to construct a model starting with building block elements and some computer aid to place and replicate these elements. The result was to build a model of the desired component or design. The second approach was to start with the component or design and break it first into topologically simple regions which in turn could be automatically broken into elements.

In parallel with this program CAD systems were evolving. These systems provided powerful means for defining complex geometry and storing it in computer readable form.

By 1978 we at Sikorsky were becoming more and more heavily involved in finite element modeling principally with manual methods. The burden of this was unacceptable. In our survey of available mesh generation systems we found no commercially available system which we felt fully exploited the available technology.

It was our opinion that finite element modeling should naturally start from defined design geometry and follow the breakdown method of mesh generation. Further we felt that the power of interactive graphics CAD systems should be fully exploited to obtain the geometry.

We had available to us some elements of the desired system. We had available an in-house developed CAD system called CAIDS (Computer Aided Interactive Design System) and a mesh generator developed at Pratt & Whitney Commercial Products Division which met most of our requirements. With these elements we were well positioned to couple them into an effective system.

IFEMS CONFIGURATION & USE

The total interactive modeling system is actually made up of three separate programs, (Figure 1). The first part is a complete computer aided design system, CAIDS, (Computer Aided Interactive Design System) modified to communicate with IFEMS. CAIDS represents a highly sophisticated interactive design system equipped with a large software library of geometry generating routines. CAIDS hardware is the ADAGE (Figure 2) vector generator terminal with real time dynamic rotation, translation and zoom display features. The CAIDS display space is three dimensional, thus allowing for geometry definition of any conceivable component. Future features will include complete three dimensional math routines capable of solving for such things as intersections of general 3D surfaces.

CAIDS uses 3D parametric rational cubic functions to store its geometry. These functions are stored as labeled groups or OBJECTS which are defined by the user. These OBJECTS become the means by which the analyst ultimately defines the finite element regions for the mesh generator and is the key to IFEMS.

The second program is IFEMS. Its purpose is to accept the component geometry from CAIDS and combine it with the user defined model attributes (number of elements, boundary conditions, loads, etc.) and ultimately prepare an input file for the mesh generator.

A functional overview of IFEMS is presented in Figure 3. The analyst begins by defining the actual component geometry using CAIDS. Upon completion of geometry definition, the analyst divides the model into regions. CAIDS then creates a neutral file which contains the model geometry grouped by OBJECTS. This neutral file is independent of the CAIDS data base and is used as a communications media to IFEMS. For large problems, the analyst may create several neutral files, the sum of which represent the complete model.

Mesh generator regions are defined by using the OBJECT feature of CAIDS. That is, a generic name is defined by the user (region number and type, i.e. 25QUAD) to identify the region. The appropriate geometry defining that region is then assigned to the object.

Phase One of IFEMS is a preprocessor which reads the neutral file(s) performing four basic functions: (1) conducts basic data validation tasks on the region data looking for such errors as missing or non-adjacent edges, (2) reformats neutral file data into in-core arrays suitable for the downstream mesh generator (See Figure 4), (3) creates additional data, (from the cubic functions) required by the mesh generator such as node points representing the corners of the regions and (4) merges in previously defined parts of the model. At the end of Phase One, a permanent file, Perm One, is created (or updated in the case of multipart models) and stored on disk. The Perm One file contains all information generated during Phase One (See Figure 5). Phase One is an interactive processor (run on a Tektronix 4014 terminal) and allows minor user intervention and minor corrections during data validation.

Phase Two Part One of IFEMS is the major section. This is where the actual finite element model region attributes are defined: element type, number of elements per region, element spacing, material definition, boundary conditions, applied load, etc. This program begins by reading the region geometry from Phase One, via the Perm One File.

The program uses a system of 40 menus to list the various options used in defining the region attributes (Figure 6). Through these menus the analyst controls the operational flow of the program. Displayed on the menu is the particular region that the analyst is working on.

The menu system is hierarchical in that a given option automatically envokes a predetermined series of lower level menus (Figure 7 & 8). This system was designed to be tutorial. Given an upper level option, such as defining the number of elements along a region edge, the program will effectively guide the analyst by prompting him for the correct input. On the other hand the user can always interrupt the flow to return to the higher level menus.

A Phase Two session would begin with the menu shown on Figure 9. The analyst will normally use Option D to define default region attributes, then Option E is used region by region to change only those attributes which are different from the default values. Changing the region attributes is accomplished by entering the letter code corresponding to the desired change. Some of these changes are made to the current menu and others will automatically involve lower level menus (See Figures 7 & 8).

The user has additional control over the program through the use of the BAIL, CONTINUE, or NEXT options. BAIL nullifies all current input and reverts to the previously defined values. CONTINUE stores all current information and returns to the previous menu for additional processing. NEXT is used to define the information for the next region or edge.

Disk data sets are used to transfer information between programs. The Perm Two file contains all the region attributes required by the mesh generator. This file is updated each time the anlayst completes a region. After all the regions attributes have been defined the program will compress the Perm Two file and apply the default values to those regions which were not explicitly accessed.

Phase Two, Part Two is basically a pre-prpcessor to the mesh generator, PWAMESH. It creates an input file (from the Perm Two file) containing the region attributes which are then read by the mesh generator. PWAMESH then generates and displays the final mesh. The analyst reviews the final mesh and passes back and forth between Part 1 and 2 until he is satisfied. At this stage, PWAMESH is instructed to generate the bulk data required by the finite element program, typically NASTRAN.

The third program, PWAMESH, is a general purpose three dimensional finite element mesh generator, developed by Pratt & Whitney Aircraft, Commercial Products Division. This program is executed from a Tektronix 4014 CRT. PWAMESH includes such features as variable thickness regions, generalized boundary condition definition, uncoupled regions for sliding mechanisms, automatic generation of quadratic temperature and pressure fields, automatic region joining as well as many other features.

PWAMESH is also equipped with an interactive graphic package for reviewing the finite element model prior to running the finite element code. Its interactive review menu (Figure 10) will plot either the input coarse region(s) or the generated fine region(s) of the model. The entire model or particular regions can be displayed at any viewing angle along with the element and node identification numbers. Offline plots can be drawn using the VERSATEC plotting system. In addition, the user can blow-up any particular section by indicating the lower left and upper right boundaries of the model. The elements of the model can also be shrunk to make individual elements more visible (Figure 11). PWAMESH will prepare a complete finite element input deck for either NASTRAN, MARC or an in-house boundary integral program.

IFEMS SPECIAL FEATURES

IFEMS was designed to minimize the amount of user defined input data. This has been achieved by allowing the user to define a set of initial model attributes. These attributes are applied to each region the first time the user displays the region. The user need only change those attributes that are unique for that region. Typically just the number of elements along an edge, or thickness.

Additional features include automatic continuous backup of user input, free field input, automatic cursor positioning and a full compliment of displayable help pages.

CONCLUDING REMARKS

IFEMS, in conjunction with CAIDS and PWAMESH provide the analyst with a completely interactive tool for modeling mechanical component independent of geometric complexity. The usual errors associated with manually preparing mesh generator input are virtually eliminated as the analyst continually 'sees' the result of his work.

Production use of IFEMS at Sikorsky has resulted in tremendous saving in both man-hours and lead time. Models that previously took three weeks are now being generated in two days. Solving complex problems and performing stress evaluation for minimum weight and cost with practical lead times has resulted in the more efficient use of the analyst's time and knowledge.

A TOTAL INTERACTIVE MODELING SYSTEM

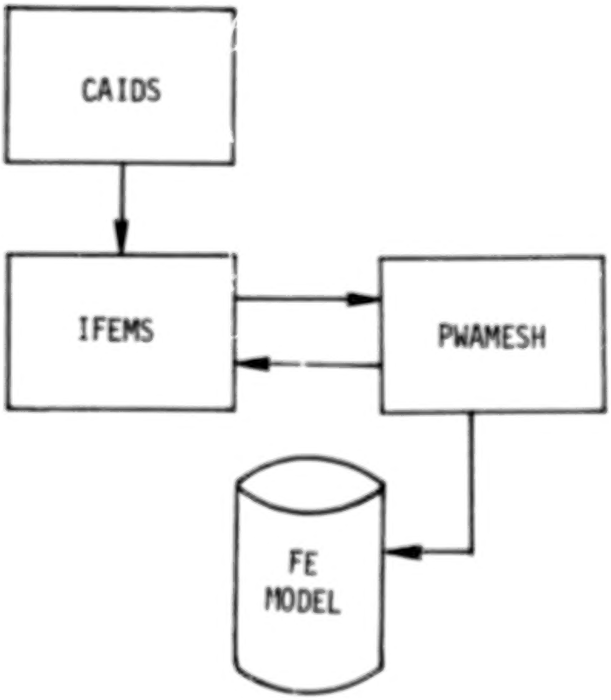

CAIDS
IFEMS
PWAMESH
FE
MODEL

FIGURE 1

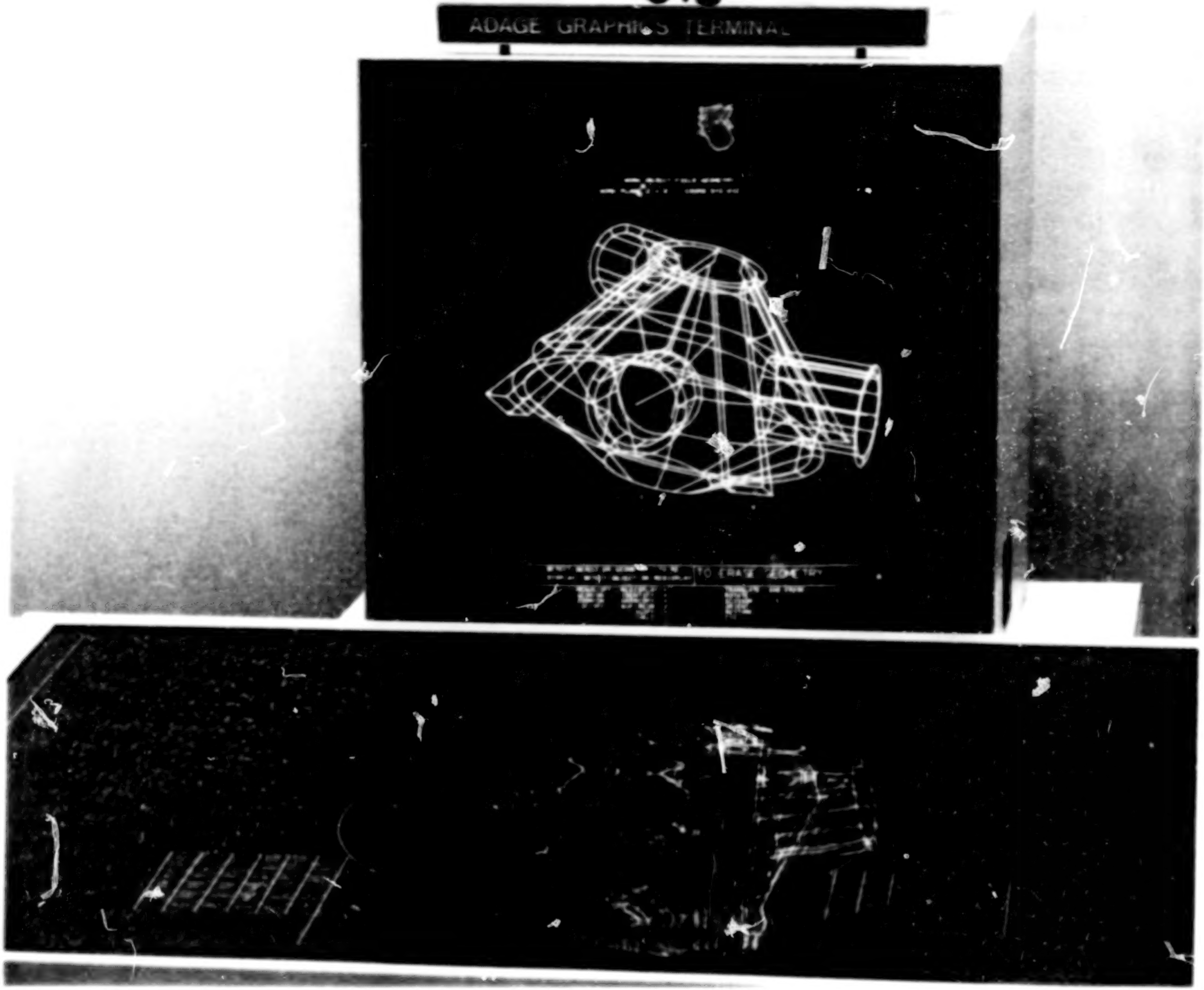

ADAGE GRAPHICS TERMINAL
TO ERASE GEOMETRY

FIGURE 2 - ADAGE Vector Generator Terminal

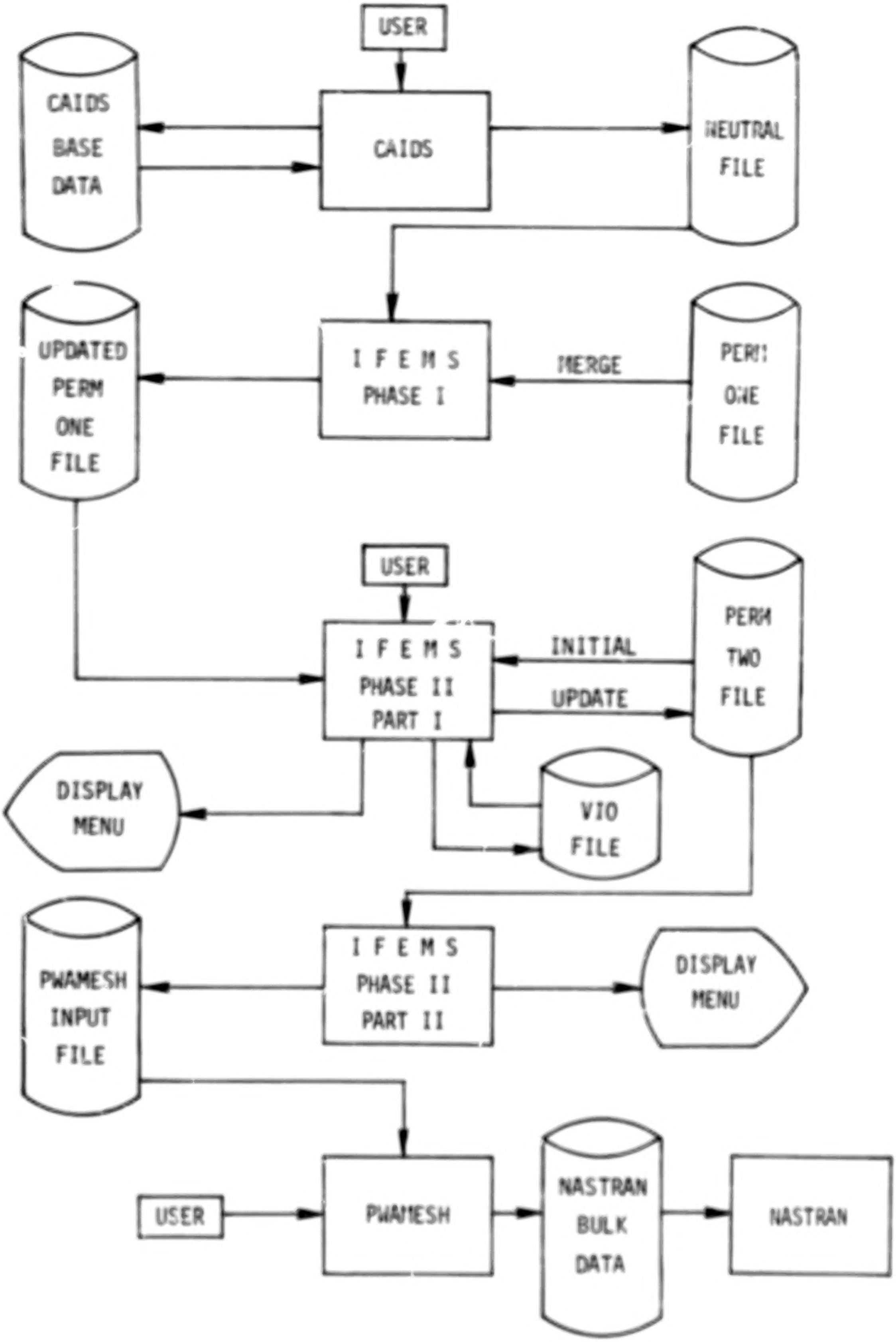

USER
CAIDS BASE DATA
CAIDS
NEUTRAL FILE
UPDATED PERM ONE FILE
IFEMS PHASE I
MERGE
PERM ONE FILE
USER
IFEMS PHASE II PART I
INITIAL
UPDATE
PERM TWO FILE
DISPLAY MENU
VIO FILE
PWAMESH INPUT FILE
IFEMS PHASE II PART II
DISPLAY MENU
USER
PWAMESH
NASTRAN BULK DATA
NASTRAN

FIGURE - 3

CONTENTS

Page

FOREWORD . . . . iii 1/A5

1. ON THE CONGRUENT FEATURE IN NASTRAN 1 1/A9
by P. R. Pamidi (Computer Sciences Corporation)

2. DEVELOPMENT OF THE LEARJET 28/29 WING USING NASTRAN ANALYSIS. 11 1/B6
by Robert J. Boroughs (Gates Learjet Corporation)

3. TENSIONING OF A BELT AROUND A DRUM USING MEMBRANE ELEMENTS 33 1/E1
by C. H. S. Chen (B. F. Goodrich Company)

4. THERMAL STRESS ANALYSIS OF SYMMETRIC SHELLS SUBJECTED TO ASYMMETRIC THERMAL LOADS 41 1/E12
by Gordon R. Negaard (ANAMET Laboratories, Inc.)

5. STRESS CONCENTRATIONS IN SCREW THREADS. 65 2/B1
by G. Peter O'Hara (U. S. Army Armament Research and Development Command)

6. CONDENSING LOADED POINTS FOR TRANSIENT BY SUBSTRUCTURING. 79 2/C6
by Thomas G. Butler (Butler Analysis)

7. IMPLEMENTATIONS OF A TRAPEZOIDAL RING ELEMENT IN NASTRAN FOR ELASTIC-PLASTIC ANALYSIS. 101 2/E5
by P. C. T. Chen and G. P. O'Hara (U. S. Army Armament Research and Development Command)

8. NORMAL MODE ANALYSIS OF THE IUS/TDRS PAYLOAD IN A PAYLOAD CANISTER/TRANSPORTER ENVIRONMENT 113 2/F5
by Karl A. Meyer (Planning Research Corporation)

9. STEADY STATE SOLUTIONS TO DYNAMICALLY LOADED PERIODIC STRUCTURES 131 2/G9
by Anthony J. Kalinowski (Naval Underwater Systems Center)

10. APPLICATIONS OF NASTRAN IN GUST RESPONSE ANALYSIS AT NORTHROP 165 3/C4
by Ashok K. Singh (Northrop Corporation)

11. IMPLEMENTATION OF NASTRAN ON THE IBM/370 CMS OPERATING SYSTEM. 189 3/F2
by Stephen S. Britten and Betsy Schumacker (M.I.T.)

MICROFILMED FROM BEST AVAILABLE COPY

Page

12. IFEMS - AN INTERACTIVE FINITE ELEMENT MODELING SYSTEM USING A CAD/CAM SYSTEM 205
by Spencer McKellip, Todd Schuman, and Spencer Lauer (Sikorsky Aircraft) 3/G5

13. APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL. 223
by James L. Rogers, Jr. (NASA Langley Research Center) 4/B6

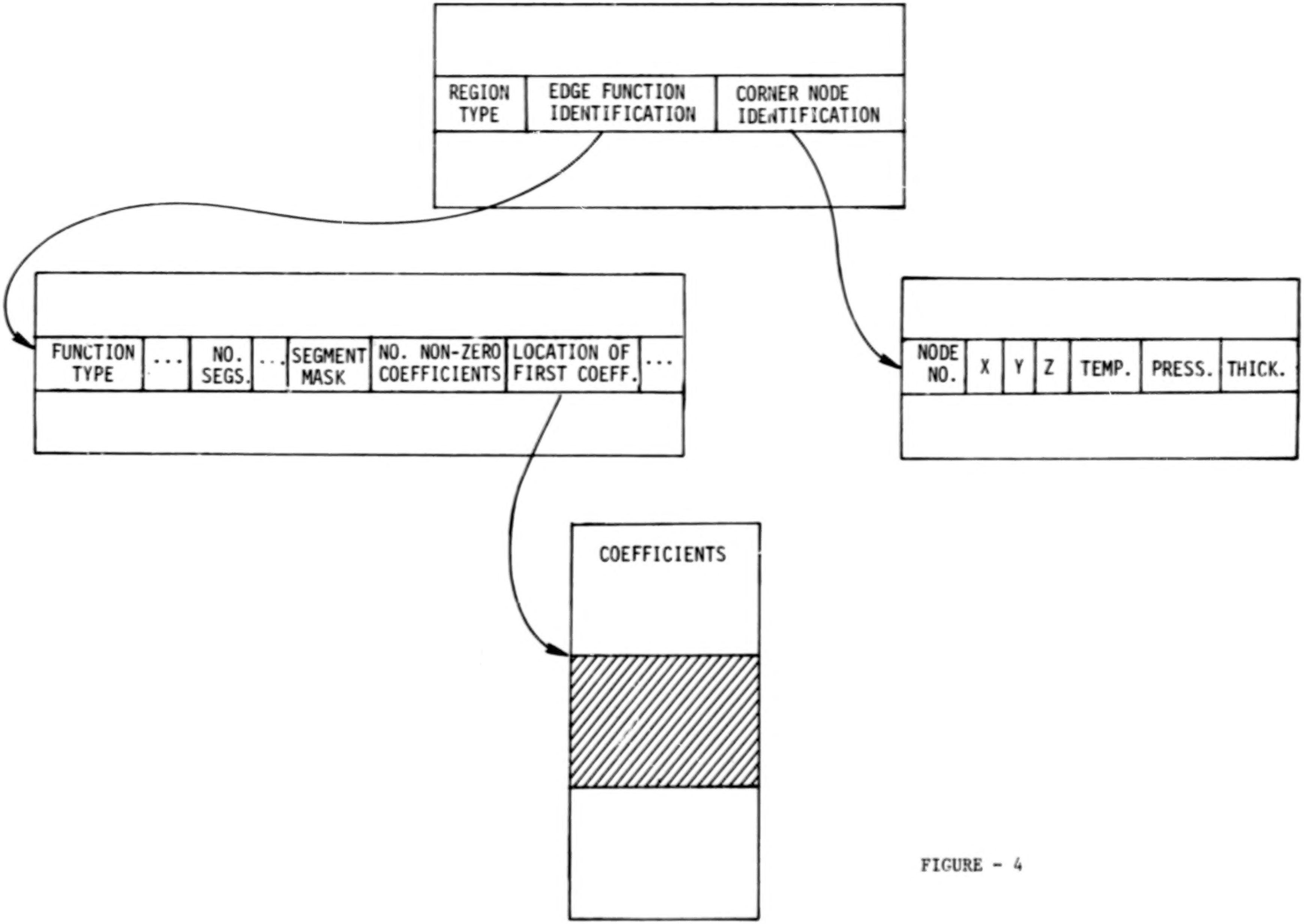

REGION TYPE
EDGE FUNCTION IDENTIFICATION
CORNER NODE IDENTIFICATION
FUNCTION TYPE
...
NO. SEGS.
...
SEGMENT MASK
NO. NON-ZERO COEFFICIENTS
LOCATION OF FIRST COEFF.
...
NODE NO.
X
Y
Z
TEMP.
PRESS.
THICK.
COEFFICIENTS

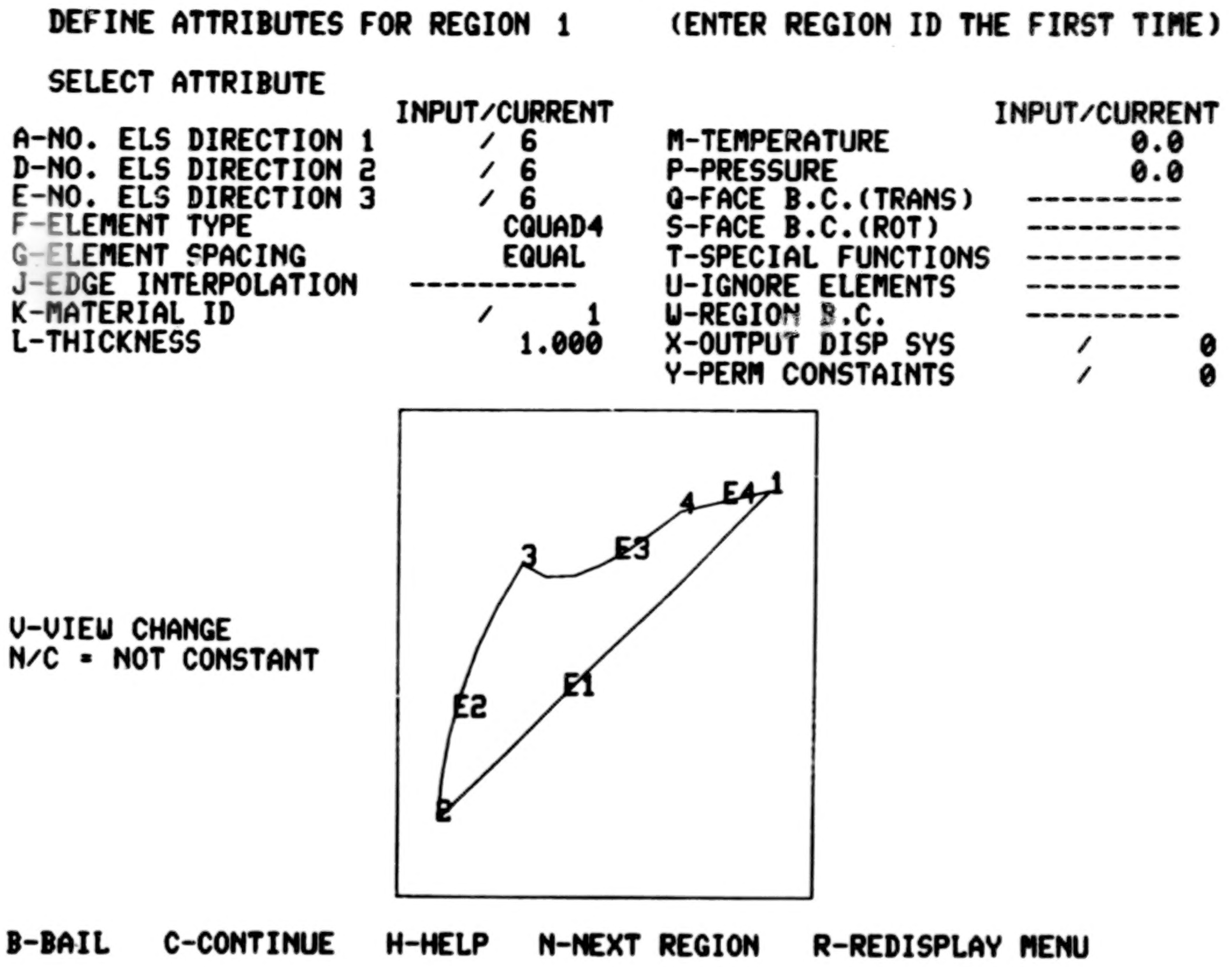
FIGURE - 4

PERM ONE FILE
STRUCTURE

- REGION DEFINITIONS
- CORNER NODE DEFINITION
- COORDINATE SYSTEM DEFINITIONS
- REGION EDGE FUNCTION DEFINITION & POINTERS
- FUNCTION COEFFICIENTS
- ARRAY PARAMETERS (ie, NO. ENTRIES, ETC.)
- READ/WRITE UNIT NUMBERS

FIGURE - 5

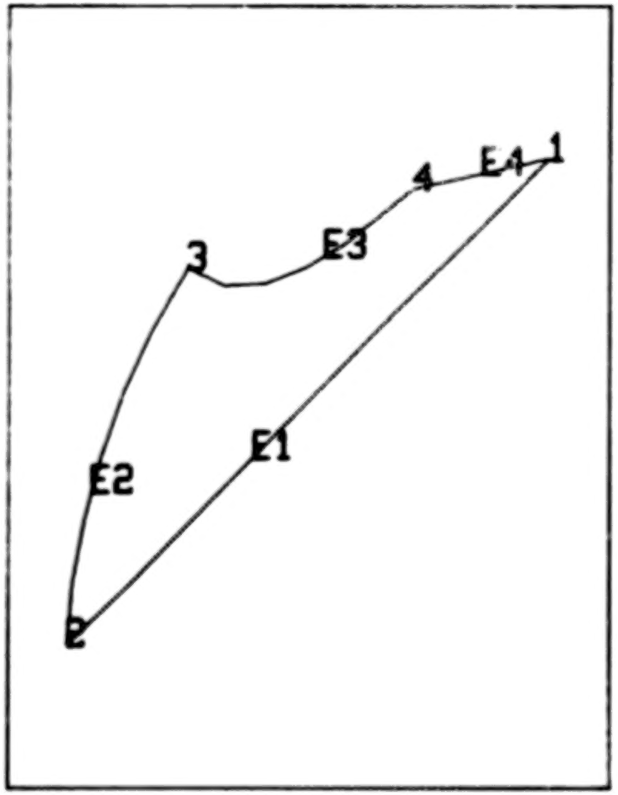
216



```
   DEFINE ATTRIBUTES FOR REGION  1      (ENTER REGION ID THE FIRST TIME)

   SELECT ATTRIBUTE
                               INPUT/CURRENT                          INPUT/CURRENT
 A-NO. ELS DIRECTION 1           /  6         M-TEMPERATURE                 0.0
 D-NO. ELS DIRECTION 2           /  6         P-PRESSURE                    0.0
 E-NO. ELS DIRECTION 3           /  6         Q-FACE B.C.(TRANS)    ----------
 F-ELEMENT TYPE                   CQUAD4      S-FACE B.C.(ROT)      ----------
 G-ELEMENT SPACING                EQUAL       T-SPECIAL FUNCTIONS   ----------
 J-EDGE INTERPOLATION    ----------           U-IGNORE ELEMENTS     ----------
 K-MATERIAL ID                   /      1     W-REGION B.C.         ----------
 L-THICKNESS                       1.000      X-OUTPUT DISP SYS        /       0
                                              Y-PERM CONSTAINTS        /       0

 V-VIEW CHANGE
 N/C = NOT CONSTANT

 B-BAIL   C-CONTINUE   H-HELP   N-NEXT REGION   R-REDISPLAY MENU
```

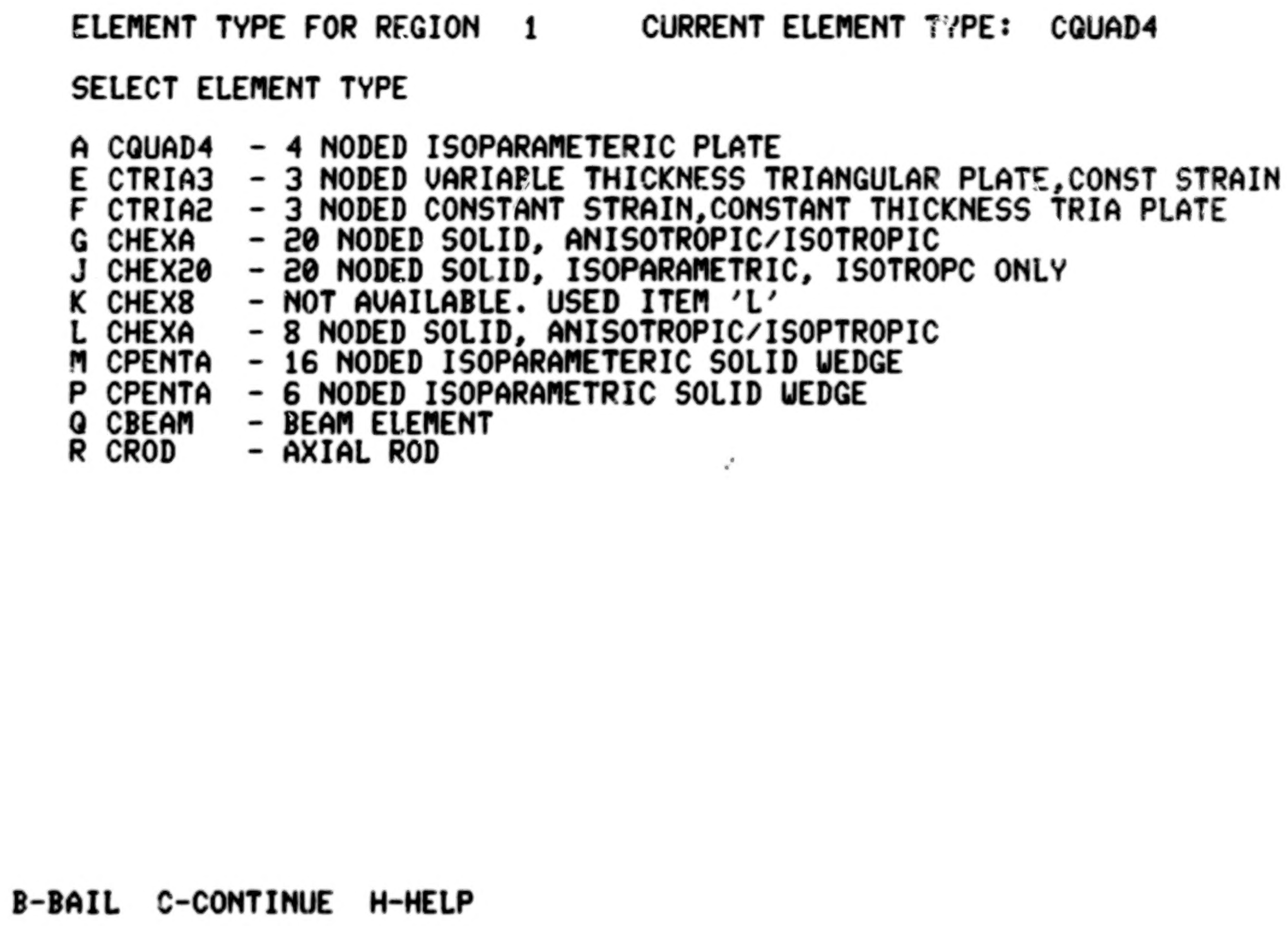
FIGURE - 6

```
ELEMENT SPACING FOR REGION   1

 QUAD REGION - PERMISSIBLE EDGES ARE
 1,2,3 OR 4

ENTER EDGE NUMBER

SELECT SPACING OPTION

A UNEQUAL
D GRADIENT
E DISPLAY CURRENT VALUES
F ENTER GRADIENT HERE
G SAME AS EDGE    OF REGION
 (NOTE: BE SURE THIS EDGE EXISTS,
  OTHERWISE PWAMESH WILL FAIL)
```



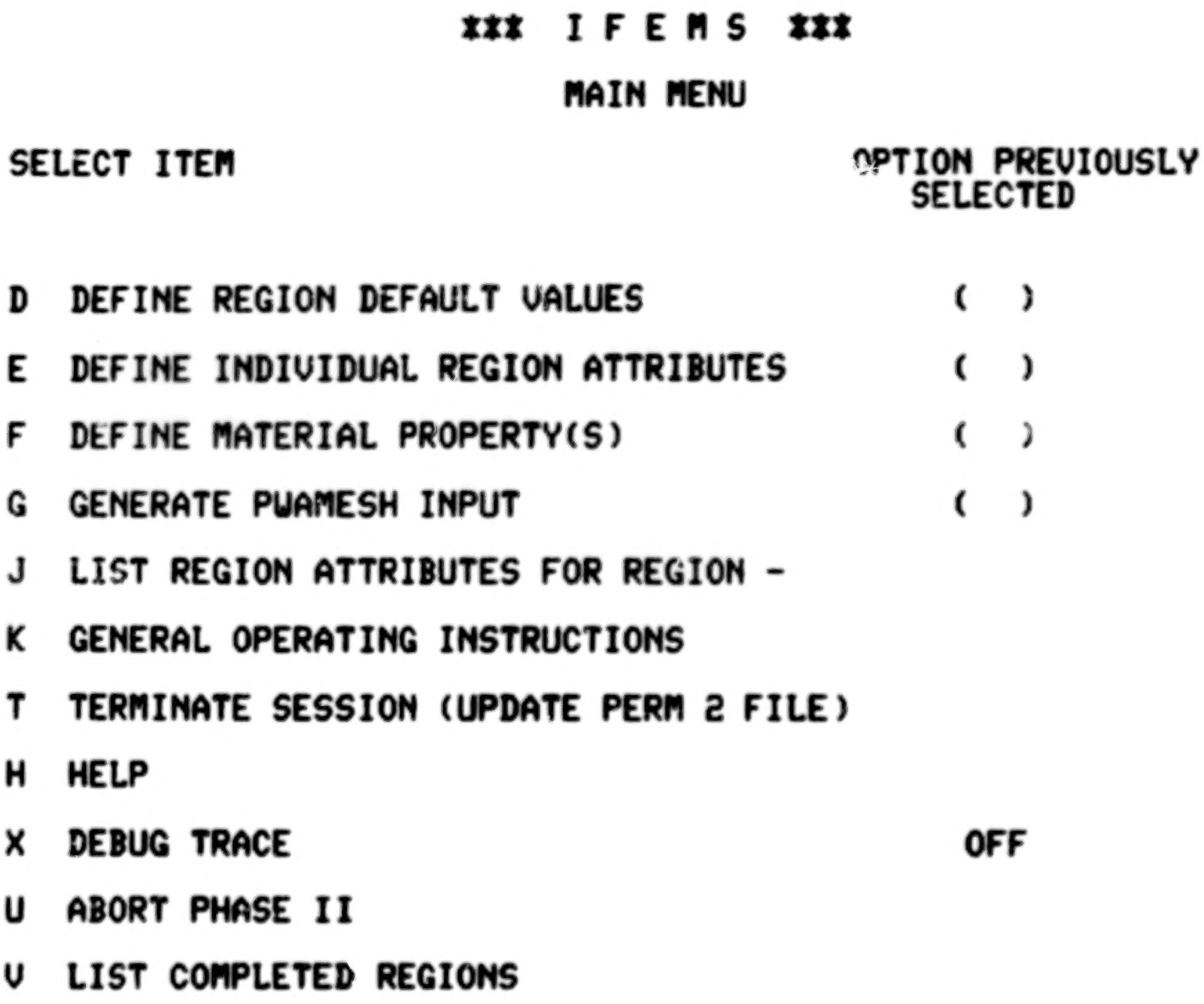
```
SELECT
B-BAIL  C-CONTINUE  V-VIEW CHANGE  H-HELP  N-NEXT EDGE
```

FIGURE - 7



```
ELEMENT TYPE FOR REGION   1     CURRENT ELEMENT TYPE:  CQUAD4

SELECT ELEMENT TYPE

A CQUAD4  - 4 NODED ISOPARAMETERIC PLATE
E CTRIA3  - 3 NODED VARIABLE THICKNESS TRIANGULAR PLATE,CONST STRAIN
F CTRIA2  - 3 NODED CONSTANT STRAIN,CONSTANT THICKNESS TRIA PLATE
G CHEXA   - 20 NODED SOLID, ANISOTROPIC/ISOTROPIC
J CHEX20  - 20 NODED SOLID, ISOPARAMETRIC, ISOTROPC ONLY
K CHEX8   - NOT AVAILABLE. USED ITEM 'L'
L CHEXA   - 8 NODED SOLID, ANISOTROPIC/ISOPTROPIC
M CPENTA  - 16 NODED ISOPARAMETERIC SOLID WEDGE
P CPENTA  - 6 NODED ISOPARAMETRIC SOLID WEDGE
Q CBEAM   - BEAM ELEMENT
R CROD    - AXIAL ROD








B-BAIL  C-CONTINUE  H-HELP
```

FIGURE - 8

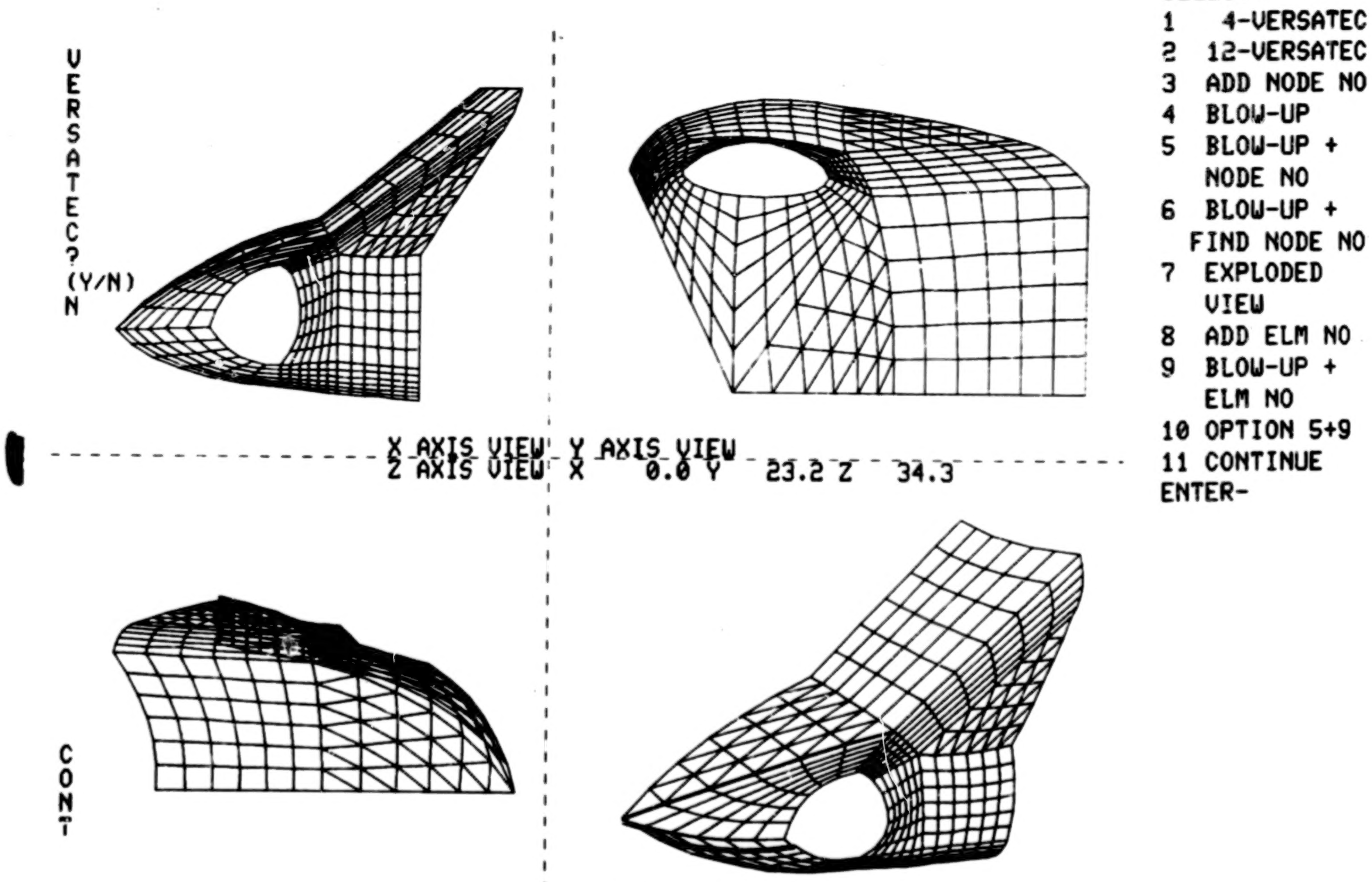


```
                          *** I F E M S ***

                              MAIN MENU

SELECT ITEM                                         OPTION PREVIOUSLY
                                                        SELECTED


D   DEFINE REGION DEFAULT VALUES                          (   )

E   DEFINE INDIVIDUAL REGION ATTRIBUTES                   (   )

F   DEFINE MATERIAL PROPERTY(S)                           (   )

G   GENERATE PWAMESH INPUT                                (   )

J   LIST REGION ATTRIBUTES FOR REGION -

K   GENERAL OPERATING INSTRUCTIONS

T   TERMINATE SESSION (UPDATE PERM 2 FILE)

H   HELP

X   DEBUG TRACE                                            OFF

U   ABORT PHASE II

V   LIST COMPLETED REGIONS
```

FIGURE - 9

BLANK PAGE

BLANK PAGE



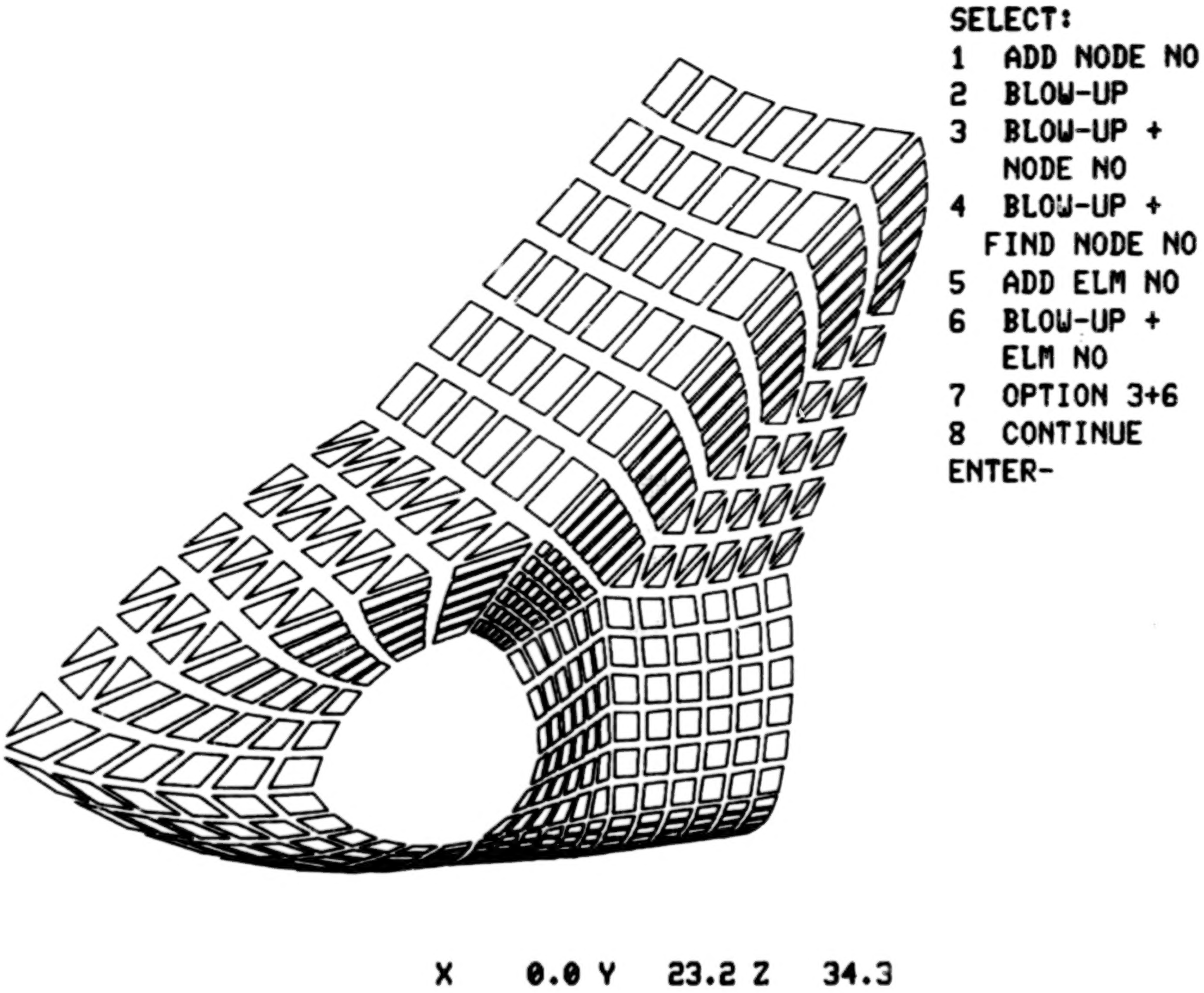
```
SELECT:
1    4-VERSATEC
2   12-VERSATEC
3   ADD NODE NO
4   BLOW-UP
5   BLOW-UP +
    NODE NO
6   BLOW-UP +
   FIND NODE NO
7   EXPLODED
    VIEW
8   ADD ELM NO
9   BLOW-UP +
    ELM NO
10  OPTION 5+9
11  CONTINUE
ENTER-
```

```
VERSATEC?(Y/N)
N
```

```
X AXIS VIEW  Y AXIS VIEW
Z AXIS VIEW  X     0.0 Y    23.2 Z   34.3
```

```
CONT
```

FIGURE - 10



```
SELECT:
1  ADD NODE NO
2  BLOW-UP
3  BLOW-UP +
   NODE NO
4  BLOW-UP +
  FIND NODE NO
5  ADD ELM NO
6  BLOW-UP +
   ELM NO
7  OPTION 3+6
8  CONTINUE
ENTER-
```

```
X     0.0 Y   23.2 Z   34.3
```

FIGURE - 11

BLANK PAGE

APPLICATION OF A DATA BASE MANAGEMENT SYSTEM TO A FINITE ELEMENT MODEL

James L. Rogers, Jr.
Langley Research Center

INTRODUCTION

In today's software market, much effort is being expended on the development of data base management systems (DBMS). Most commercially available DBMS were designed for business use. However, the need for such systems within the engineering and scientific communities is becoming apparent.

A potential DBMS application that appears attractive is the handling of data for finite element engineering models. The purpose of this paper is to explore the application of a commercially available, business-oriented DBMS to a structural engineering, finite element model. The model, DBMS, an approach to using the DBMS, advantages and disadvantages are explored in detail; and plans for research on a scientific and engineering DBMS are discussed.

THE FINITE ELEMENT MODEL

Many organizations use more than one finite element computer program for analyzing structural models. For instance NASTRAN might be used because of its generality, standardization, and universal acceptance, or SPAR might be used to gain quick turnaround and interactive capability. Input into these programs usually consists of data representing a finite element model. Typically, the model has a single physical representation which can easily be pictured as a hierarchial structure composed of the complete model, substructures, elements, and grid points (see fig. 1). This single representation, however, can take the form of input to any one or more of the many finite element programs available today.

If an engineer wishes to use more than one application program for analysis, a separate input deck must be maintained for each program, although in truth only one model is being analyzed. This proliferation of input decks can lead to a loss of integrity in the data, because of the difficulty in keeping all related input decks at the same level of modification -- especially when many changes are being made to the model. To solve the proliferation problem and maintain integrity of the data, the finite element model input data can be stored in a data base.

THE DATA BASE MANAGEMENT SYSTEM[1]

Storing the model input data in a data base independent of all the application programs that are to make use of it as input can solve the problem of proliferation of input decks and aid in maintaining the integrity of the model. The data base is loaded into the DBMS in the steps shown in figure 2. Step 1 simply defines a data base name for the user to work with. The model input data must now be stored in data elements, the basic components of a data base. By using the data base definition the hierarchial structure of the finite element model is easily adapted to the hierarchial structure of the data base. A typical data base definition is shown in figure 3. Particular attention should be paid to its generality and its hierarchial arrangement. Most any finite element model, large or small, can be input with this definition. The hierarchial structure can be seen by scanning the indentation of the definition. Data element 1 allows input of a model name. Data element 2 is a repeating group (RG) of substructures. A repeating group allows more than one substructure per model to be input. Data element 3 allows input of a substructure name. Data element 4 is a repeating group of finite elements within a substructure. Data elements 5-10 provide for input of element information. Data element 11 is a repeating group of grid points within a finite element. Data elements 12-16 provide for input of grid point information. Data elements 20-49 define supporting information (not necessarily hierarchial) about the model such as element properties, single point constraints, omitted coordinates, and eigenvalue information. Step 3 (fig. 2) inputs the finite element model into the data base through the data base definition.

After the data has been loaded into the data base, it cannot be used in an application program without undergoing some sort of conversion. The easiest approach when working with already existing programs, is to use FORTRAN pre-processor programs to convert the data from the independent data base format to an input format for a particular application program such as NASTRAN or SPAR (see fig. 4). Pre-processor programs for converting data base input into NASTRAN and SPAR input decks are shown in Appendices I and II, respectively. A different pre-processor is needed for each application program that will use the model data as input. A data manipulation language (DML) which can be embedded within the pre-processor programs allows the user to write a FORTRAN program that will interact with the data base. The DML statements of the programs in the appendices have *PL in columns 1-3.

---

[1] SYSTEM 2000, a commercially available data base management system developed by MRI Systems Corporation, was used for all research done for this paper. However, almost any DBMS with a FORTRAN procedural language and query language could have been used. Use of commercial products and names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers either expressed or implied, by the National Aeronautics and Space Administration.

Another powerful tool of some DBMS is the query language. This language allows the user to interactively query the data base. Using this language, he can list the grid points associated with a particular finite element or vice-versa he can list the finite elements associated with a particular grid point. Many other associations can be listed depending upon the relationships defined in the data base definition.

This has only been a brief introduction into the world of DBMS. More details concerning the data base definition, data base input, DML, query language, and creation of a data base, can be obtained from manuals available from the developer of a specific DBMS.

APPLILCATION OF A DBMS TO A FINITE ELEMENT CANTILEVER BEAM

The data base input of a cantilever beam is shown in figure 5. Close attention should be paid to the correlation of numbers between figure 3 (data base definition) and figure 5 (data base input). To demonstrate the process shown in figure 4, a cantilever beam finite element model is created (see fig. 6). The data base definition and data base input described above and shown in figures 3 and 5 are used in this demonstration. After the data is stored in the data base, it is input into the pre-processor programs shown in appendices. Other input that does not pertain to the model but is necessary input to NASTRAN and SPAR (e.g., Executive and Case Control decks of NASTRAN) are read in from another file and inserted into the proper place in the output from the pre-processor programs. The pre-processor program for NASTRAN is not set up to handle PARAM and CNGRNT Bulk Data cards, thus these cards are also read from the same file as the Executive and Case Control cards. The pre-processor programs could be expanded and made sufficiently general to handle almost any input requirement. The pre-processor programs output a file ready for input into either NASTRAN (fig. 7) or SPAR (fig. 8). Thus it is shown that a finite element model data can be stored in a data base, independent of any application program which would use the model data as input. With the aid of pre-processor FORTRAN programs, the model data can subsequently be converted to a format for direct input into one of several application programs.

ADVANTAGES AND DISADVANTAGES

The advantages in using a data base to store a finite element model have been discussed in detail in previous sections so only a summary appears in this section. The data base requires only one copy of the model to be stored even though the data may be used as input to more than one application program. This situation aids in maintaining the integrity of the model, especially a model undergoing frequent changes. The stored data are independent of the application programs that use it, thus changes can be made to the programs and/or the data without having cross effects. A query language in the DBMS gives the user easy interactive access to the data, allowing listings of many types of data elements and cross-references between data elements.

The use of a DBMS is not without disadvantages. Of primary importance is the computer storage overhead (disk space) required to store a model. The small cantilever beam problem described earlier took almost $11{,}000_{10}$ words of storage. Changing the NON-KEY values to KEY in the data base definition results in the data base requiring $23{,}500_{10}$ words of storage. (A KEY value permits the user to make sorts on that particular data element.) The user must judiciously choose which data elements are KEY to minimize storage overhead. Another disadvantage is that most business-oriented DBMS do not permit "E" formats (scientific notation). Since many structural terms involve very large or very small numbers, the "E" format is essential. One final disadvantage is the lack of support for matrix or vector input to the data base. Although this capability is not required to store finit element model data, it would be particularly useful when storing and querying intermediate or final results from the application programs.

These are the major advantages and disadvantages revealed during the course of the research. The user will have to make trade-offs in determining the most optimal way to use a DBMS with finite element models and computer programs.

PLANS FOR DBMS WORK WITH FINITE ELEMENT MODELS

New data base management systems (DBMS) are now being designed and developed for engineering and scientific applications. For this reason, as well as the previously discussed disadvantages, no future research is planned using a commercial business-oriented DBMS with a finite element model. Future research is planned using E/S DMS, a scientific and engineering oriented DBMS being developed under a NASA grant at the University of Texas. E/S DMS is scheduled to be delivered to NASA Langley in August 1979. It executes on CDC CYBER computers with a NOS operating system. It has been proposed to implement E/S DMS on a minicomputer allowing users to distribute the data base as well as the processing between a mainframe and a minicomputer. This arrangement would give the user a very powerful tool; because the design and modeling could then be done on the faster (line speed) interactive graphics terminal, while the "number crunching" applications programs such as NASTRAN and SPAR could be executed on the more powerful mainframe. Future research is planned along these lines.

CONCLUDING REMARKS

A structural engineering, finite element model can be represented as a hierarchial structure within a business-oriented data base management system (DBMS). Use of the DBMS and conversion pre-processor computer programs, allow the data to be stored independent of application programs that use the data as input. This arrangement has the advantage that only one copy of the data needs to be stored and maintained even though more than one application program can use the data as input. The DBMS may also allow the user to query the data base as an aid in his modeling. The data manipulation language of the DBMS gives the application programmer the power and flexibility of FORTRAN combined with the data storage capabilities of the data base. A small example problem demonstrates the use of a data base with one specific finite element model. Output of the conversion programs is shown in the form of input decks to both the NASTRAN and SPAR structural analysis application programs. Advantages and disadvantages are described to show that while although a DBMS created for business-oriented data can be useful for handling a finite element engineering model, it does not have all the capabilities required for engineering and scientific computing.

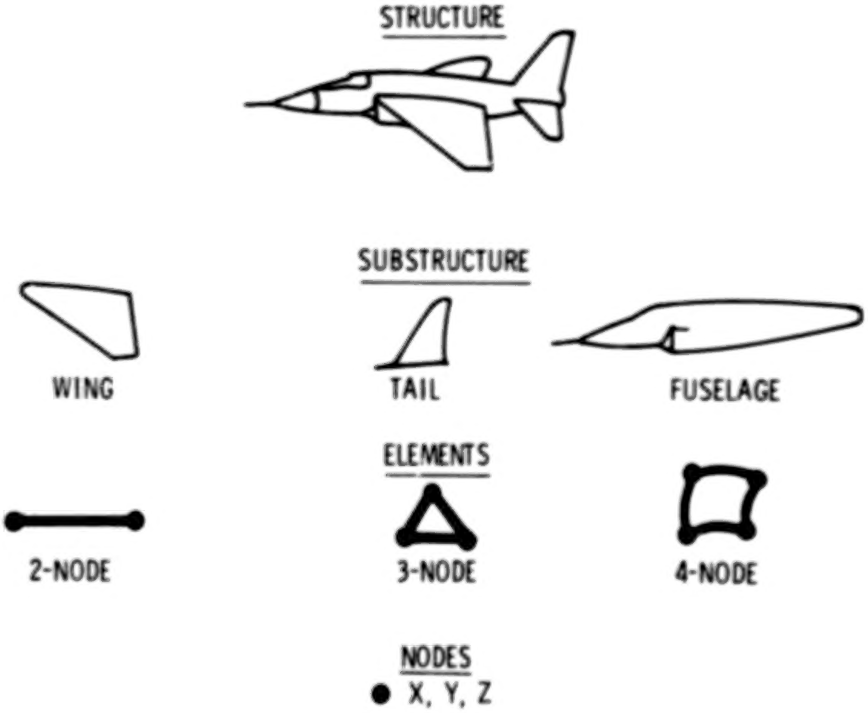

STRUCTURE
SUBSTRUCTURE
WING
TAIL
FUSELAGE
ELEMENTS
2-NODE
3-NODE
4-NODE
NODES
● X, Y, Z

Figure 1.- Hierarchial structure of a finite element model.

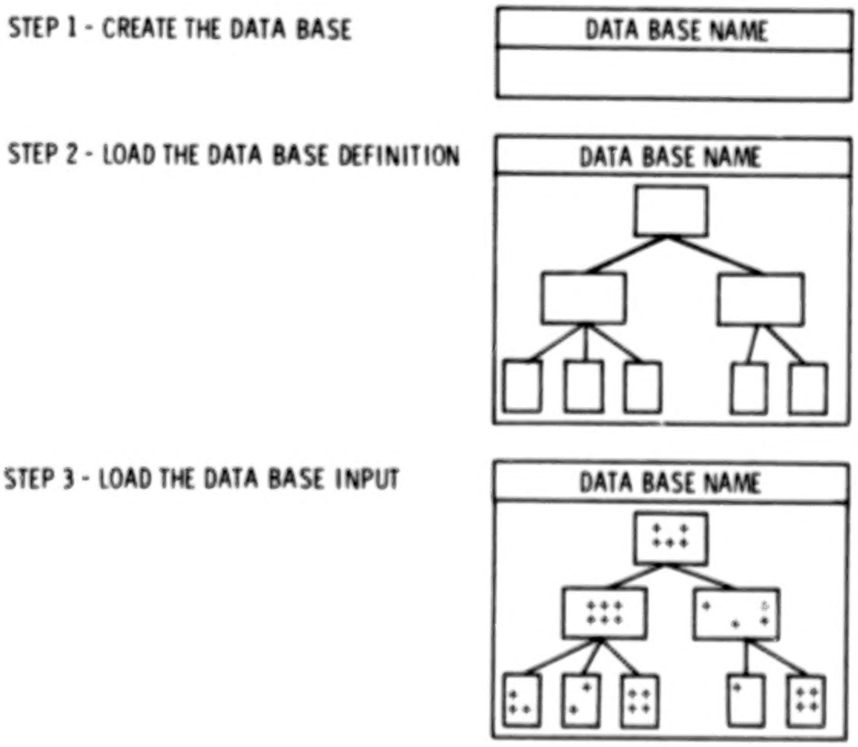

STEP 1 - CREATE THE DATA BASE
DATA BASE NAME
STEP 2 - LOAD THE DATA BASE DEFINITION
DATA BASE NAME
STEP 3 - LOAD THE DATA BASE INPUT
DATA BASE NAME

Figure 2.- Steps for loading a data base.

```
1* MODEL (NAME X(21)):
2* SUBSTRUCTURES(RG):
  3* SUBSTRUCTURE NAME (NAME X(10) IN 2):
  4* SUBSTRUCTURE ELEMENTS(RG IN 2):
    5* ELEMENT ID (INTEGER NUMBER 9(7) IN 4):
    6* ELEMENT NAME (NON-KEY NAME X(7) IN 4):
    7* X1 VECTOR COMPONENT (NON-KEY DECIMAL NUMBER IN 4):
    8* X2 VECTOR COMPONENT (NON-KEY DECIMAL NUMBER IN 4):
    9* X3 VECTOR COMPONENT (NON-KEY DECIMAL NUMBER IN 4):
    10* ELEMENT PROPERTY ID (INTEGER NUMBER 9(7) IN 4):
    11* ELEMENT GRID POINTS (RG IN 4):
       12* GRID POINT ID (INTEGER NUMBER 9(7) IN 11):
       13* X COORDINATE (NON-KEY DECIMAL NUMBER IN 11):
       14* Y COORDINATE (NON-KEY DECIMAL NUMBER IN 11):
       15* Z COORDINATE (NON-KEY DECIMAL NUMBER IN 11):
       16* PERMANENT SPC (NON-KEY INTEGER NUMBER 9(7) IN 11):
  20* ELEMENT PROPERTIES (RG IN 2):
    21* PROPERTY ID (INTEGER NUMBER 9(7) IN 20):
    22* MATERIAL ID (NON-KEY INTEGER NUMBER 9(7) IN 20):
    23* AREA (NON-KEY NAME X(7) IN 20):
    24* MOMENTS OF INERTIA (NON-KEY NAME X(7) IN 20):
    25* YOUNGS MODULUS (NON-KEY NAME X(7) IN 20):
    26* SHEAR MODULUS (NON-KEY DECIMAL NUMBER IN 20):
    27* POISSONS RATIO (NON-KEY DECIMAL NUMBER IN 20):
    28* MASS DENSITY (NON-KEY DECIMAL NUMBER IN 20):
    29* PROPERTY NAME (NON-KEY NAME X(7) IN 20):
  35* SINGLE POINT CONSTRAINTS(RG IN 2):
    36* SPC SET ID (NON-KEY INTEGER NUMBER 9(7) IN 35):
    37* SPC GRID ID (NON-KEY INTEGER NUMBER 9(7) IN 35):
    38* SPC COMPONENT NUMBER (NON-KEY INTEGER NUMBER 9(7) IN 35):
  39* OMITTED COORDINATES(RG IN 2):
    40* OMIT GRID ID (NON-KEY INTEGER NUMBER 9(7) IN 39):
    41* OMIT COMPONENT NUMBER (NON-KEY INTEGER NUMBER 9(7) IN 39):
  42* EIGENVALUE INFORMATION(RG IN 2):
    43* EIGENVALUE SET ID (NON-KEY INTEGER NUMBER 9(7) IN 42):
    44* METHOD (NON-KEY NAME X(7) IN 42):
    45* LO FREQ (NON-KEY DECIMAL NUMBER IN 42):
    46* HI FREQ (NON-KEY DECIMAL NUMBER IN 42):
    47* ESTIMATE OF ROOTS (NON-KEY INTEGER NUMBER 9(7) IN 42):
    48* DESIRED ROOTS (NON-KEY INTEGER NUMBER 9(7) IN 42):
    49* NORMALIZING METHOD (NON-KEY NAME X(7) IN 42):
```

Figure 3.- Data base definition for a finite element model.

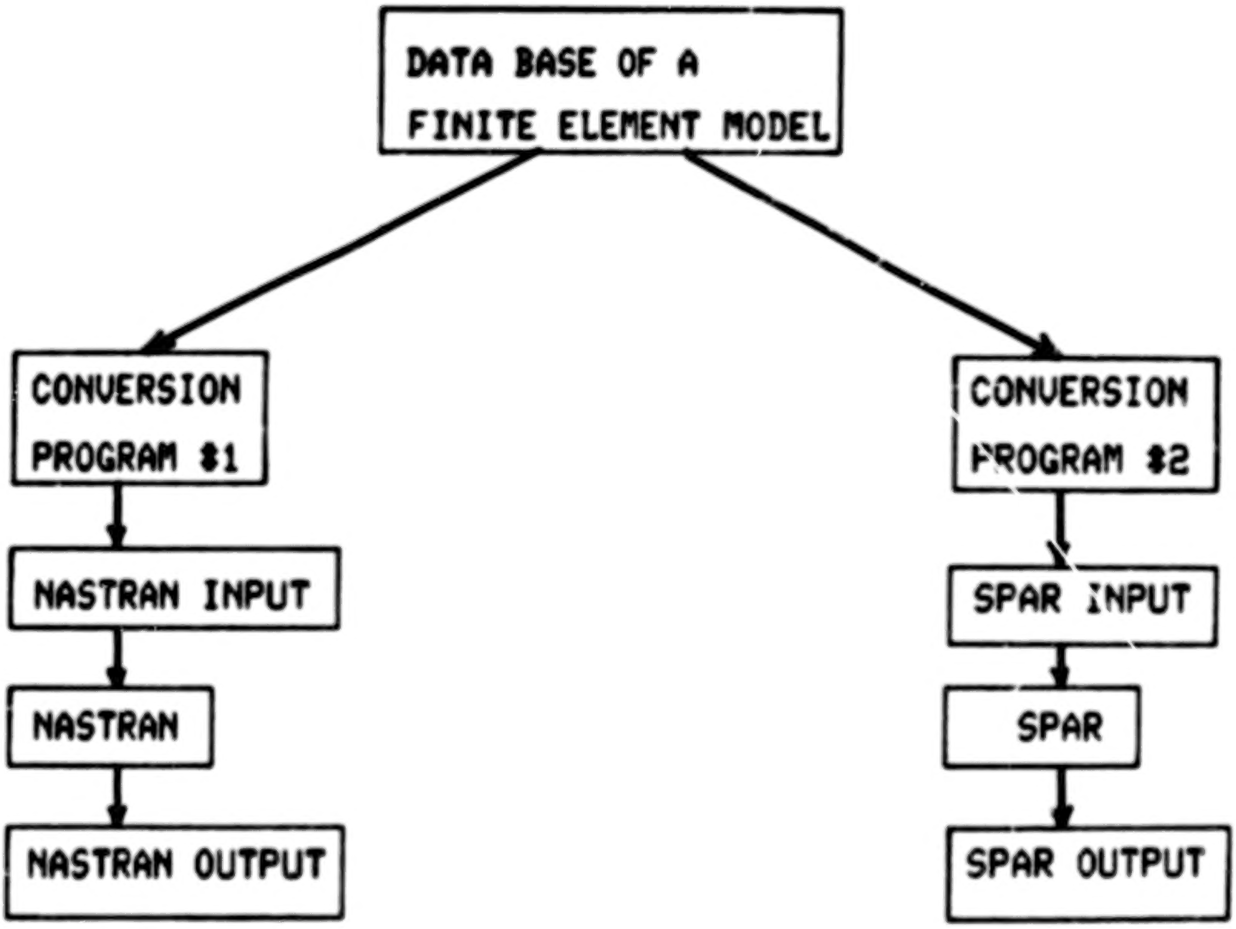

DATA BASE OF A FINITE ELEMENT MODEL
CONVERSION PROGRAM #1
NASTRAN INPUT
NASTRAN
NASTRAN OUTPUT
CONVERSION PROGRAM #2
SPAR INPUT
SPAR
SPAR OUTPUT

Figure 4.- Flowchart of finite element data base process.

```
1*DATA BASE TEST*
2*3*CANT BEAM*
 4*5*11*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*1*   13*0,0*  16*1345*
 11*12*2*
4*5*12*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*2*  13*0,1*  16*1345*
 11*12*3*
4*5*13*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*3*   13*0,2*  16*1345*
 11*12*4*
4*5*14*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*4*    13*0,3*  16*1345*
 11*12*5*
4*5*15*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*5*   13*0,4*  16*1345*
 11*12*6*
4*5*16*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*6*   13*0,5*  16*1345*
 11*12*7*
4*5*17*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*7*   13*0,6*  16*1345*
 11*12*8*
4*5*18*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*8*   13*0,7*  16*1345*
 11*12*9*
4*5*19*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*9*   13*0,8*  16*1345*
 11*12*10*
4*5*20*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*10*  13*0,9*  16*1345*
 11*12*11*
4*5*21*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*11*  13*1,0*  16*1345*
 11*12*12*
4*5*22*  6*CBAR*  7*0,*  8*1,*  9*0,*  10*201*
 11*12*12*  13*1,1*  16*1345*
 11*12*13*  13*1,2*  16*1345*
20*21*201*
   22*6*
   23*6,0-4*
   24*5,0-9*
   25*7,0+10*
   27*,3*
   28*86*
   29*PBAR*
35*36*20*
   37*1*
   38*26*
39*40*13*     41*6*
39*40*2*      41*6*
39*40*3*      41*6*
39*40*4*      41*6*
39*40*5*      41*6*
39*40*6*      41*6*
39*40*7*      41*6*
39*40*8*      41*6*
39*40*9*      41*6*
39*40*10*     41*6*
39*40*11*     41*6*
39*40*12*     41*6*
42*43*12*
   44*INV*
   45*0,0*
   46*500,*
   47*3*
   48*3*
   49*MAX*
END**
```

Figure 5.- Data base input for a cantilever beam.

MICROFILMED FROM
BEST AVAILABLE COPY

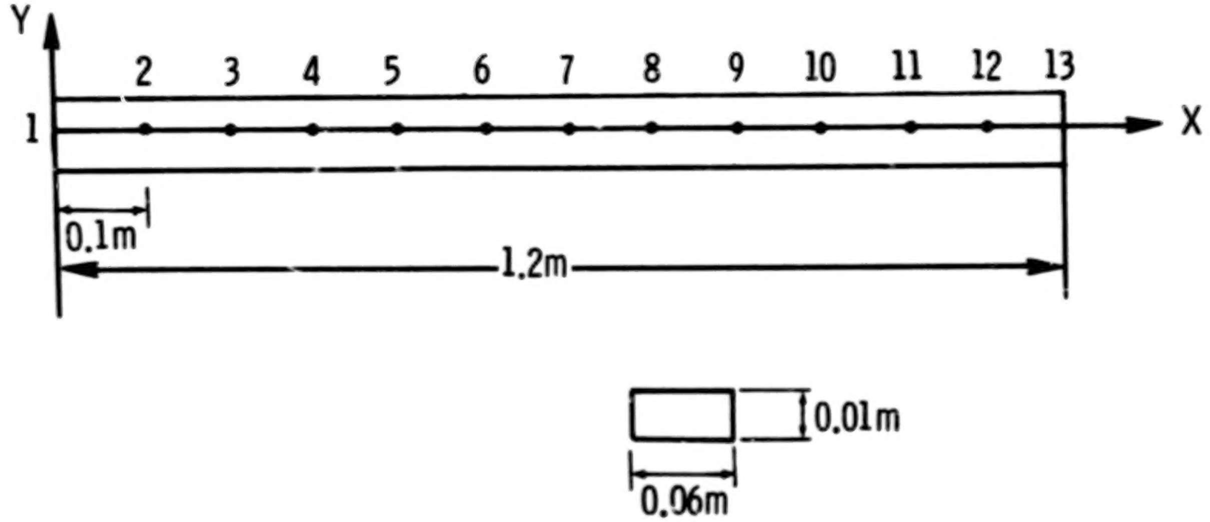
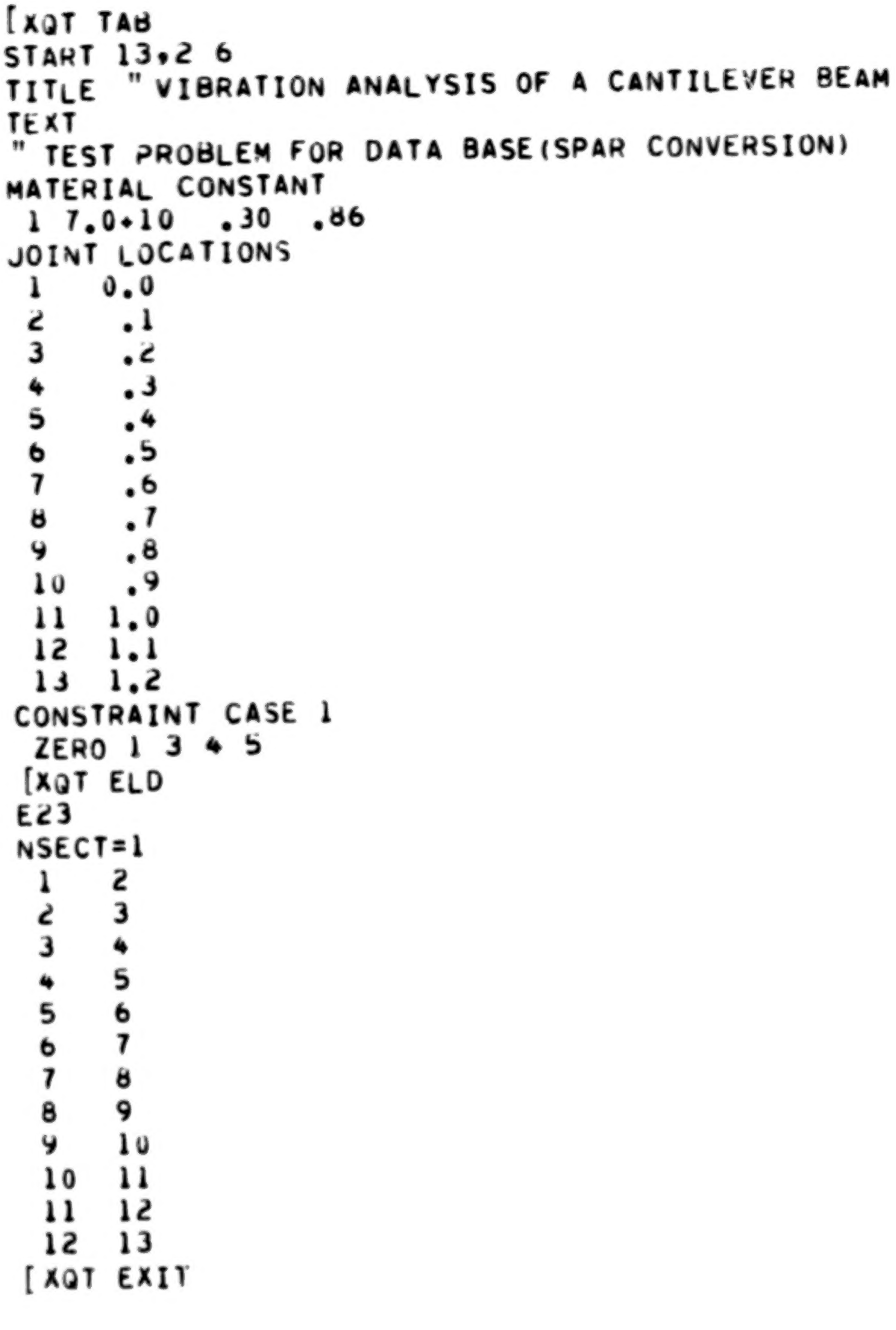

Y
X
1
2
3
4
5
6
7
8
9
10
11
12
13
0.1m
1.2m
0.01m
0.06m

Figure 6.- Cantilever beam finite element model.

```
ID DATABASE,TEST
APP DISP
SOL 3,0
TIME 4
CEND
TITLE = TEST PROBLEM FOR DATA BASE(NASTRAN CONVERSION)
SUBTITLE = VIBRATION ANALYSIS OF A CANTILEVER BEAM
  LINE = 40
  ECHO = BOTH
  METHOD = 12
  SPC = 20
OUTPUT
  DISP = ALL
BEGIN BULK
CBAR              11       201         1         2       0.0       1.0       0.0         1
CBAR              12       201         2         3       0.0       1.0       0.0         1
CBAR              13       201         3         4       0.0       1.0       0.0         1
CBAR              14       201         4         5       0.0       1.0       0.0         1
CBAR              15       201         5         6       0.0       1.0       0.0         1
CBAR              16       201         6         7       0.0       1.0       0.0         1
CBAR              17       201         7         8       0.0       1.0       0.0         1
CBAR              18       201         8         9       0.0       1.0       0.0         1
CBAR              19       201         9        10       0.0       1.0       0.0         1
CBAR              20       201        10        11       0.0       1.0       0.0         1
CBAR              21       201        11        12       0.0       1.0       0.0         1
CBAR              22       201        12        13       0.0       1.0       0.0         1
GRID               1                0.00                                    1345
GRID               2                 .10                                    1345
GRID               3                 .20                                    1345
GRID               4                 .30                                    1345
GRID               5                 .40                                    1345
GRID               6                 .50                                    1345
GRID               7                 .60                                    1345
GRID               8                 .70                                    1345
GRID               9                 .80                                    1345
GRID              10                 .90                                    1345
GRID              11                1.00                                    1345
GRID              12                1.10                                    1345
GRID              13                1.20                                    1345
PBAR             201         6     6.0-4     5.0-9
MAT1               6    7.0+10                 0.3       .86
OMIT              13         6
OMIT               2         6
OMIT               3         6
OMIT               4         6
OMIT               5         6
OMIT               6         6
OMIT               7         6
OMIT               8         6
OMIT               9         6
OMIT              10         6
OMIT              11         6
OMIT              12         6
SPC               20         1        26
EIGR              12INV              0.0     500.0         3         3                   +BC
+BC       MAX
PARAM     COUPMASS           1
CNGRNT            11      12THRU                22
ENDDATA
```

Figure 7.- NASTRAN input deck output from pre-processor program NTRNS2K.



```
[XQT TAB
START 13,2 6
TITLE "VIBRATION ANALYSIS OF A CANTILEVER BEAM
TEXT
" TEST PROBLEM FOR DATA BASE(SPAR CONVERSION)
MATERIAL CONSTANT
 1 7.0+10  .30  .86
JOINT LOCATIONS
 1   0.0
 2    .1
 3    .2
 4    .3
 5    .4
 6    .5
 7    .6
 8    .7
 9    .8
 10   .9
 11  1.0
 12  1.1
 13  1.2
 CONSTRAINT CASE 1
  ZERO 1 3 4 5
 [XQT ELD
 E23
 NSECT=1
  1   2
  2   3
  3   4
  4   5
  5   6
  6   7
  7   8
  8   9
  9   10
  10  11
  11  12
  12  13
  [XQT EXIT
```

Figure 8.- SPAR input deck output from pre-processor program SPARS2K.

APPENDIX A

PRE-PROCESSOR PROGRAM TO CONVERT DATA IN DATA BASE TO NASTRAN INPUT FORMAT

```
      PROGRAM NTRNS2K(INPUT,OUTPUT,TAPE10=OUTPUT,TAPE9)
C
C  THE PURPOSE OF THE PROGRAM IS TO CONVERT DATA FROM A DATA BASE
C          INTO NASTRAN BULK DATA FORMAT
C
      INTEGER GRID(30),PSPC(30),PID(10),C5,C10,C12,C21,C22,C16
      INTEGER C36,C37,C40,C41,C43,C47,C48
      INTEGER SCHNME,RCODE,FILLER,PASSWD,DATASN,DATASP,TIME,DATE,CYCLE,
     1  SEPSYM,ENTRYT,STATUS
C
C  COMMON BLOCK FOR PICASSO DATA BASE
C
*PL   COMMBLOCK/PICASSO/SCHNME,RCODE,FILLER,LASTDS,PASSWD,DATASN,DATASP,
*PL   LEVEL,TIME,DATE,CYCLE,SEPSYM,ENTRYT,STATUS.
C
C  SCHEMA FOR TOTAL STRUCTURE
C
*PL   SCHEMA/LMONE OF PICASSO/C1.
C
C  SCHEMA FOR REPEATING GROUP 2 (SUBSTRUCTURE NAMES)
C
*PL   SCHEMA/LZERO OF PICASSO/C3.
C
C  SCHEMA FOR REPEATING GROUP 4 (ELEMENTS)
C
*PL   SCHEMA/LONE OF PICASSO/C5,C6,C7,C8,C9,C10.
C
C  SCHEMA FOR REPEATING GROUP 11 (GRID POINTS)
C
*PL   SCHEMA/LTWO OF PICASSO/C12,C13,C14,C15,C16.
C
C  SCHEMA FOR REPEATING GROUP 20 (ELEMENT PROPERTIES)
C
*PL   SCHEMA/LTHRE OF PICASSO/C21,C22,C23,C24,C25,C26,C27,C28,C29.
C
C  SCHEMA FOR REPEATING GROUP 36 (SINGLE POINT CONSTRAINTS)
C
*PL   SCHEMA/LFOR OF PICASSO/C36,C37,C38.
C
C  SCHEMA FOR REPEATING GROUP 39 (OMITTED COORDINATES)
C
*PL   SCHEMA/LFIVE OF PICASSO/C40,C41.
C
C  SCHEMA FOR REPEATING GROUP 42 (EIGENVALUE INFORMATION)
C
*PL   SCHEMA/LATE OF PICASSO/C43,C44,C45,C46,C47,C48,C49.
*PL   END SCHEMAS.
      DIMENSION DECKIN(8) , XCORD(30) , YCORD(30) , ZCORD(30)
      DATA BEGBULK/10HBEGIN BULK/
C
C  FIRST READ IN THE EXECUTIVE AND CASE CONTROL DECKS AND STORE ON
C                                TAPE10
   1  READ (9,2)(DECKIN(I),I=1,8)
   2  FORMAT(8A10)
      WRITE(10,2) (DECKIN(I),I=1,8)
      IF(DECKIN(1).NE.BEGBULK) GO TO 1
```

```
C
*PL    START S2K.
       PASSWD=6HROGERS
C
C  OPEN PICASSO DATA BASE
C
*PL    OPEN PICASSO.
C
C  CHECK RETURN CODE
C
       IF(RCODE.EQ.0) GO TO 10
       PRINT 5,RCODE
    5  FORMAT(* DATA BASE CAN NOT BE OPENED - RETURN CODE = *,I3)
       GO TO 900
C
C  CHECK STATUS
C
   10  IF(STATUS.EQ.0) GO TO 20
       PRINT 15,STATUS
   15  FORMAT(* DATA BASE DAMAGED - STATUS = *,I2)
       GO TO 900
C
C
C  CREATE ELEMENT CARDS
C
   20  READ(9,25) C3
   25  FORMAT(A10)
       IF(EOF(9)) 800,30
   30  CONTINUE
       I=0
       M=0
C
C  RETRIEVE SUBSTRUCTURE FROM DATA BASE
C
C
*PL    GET1 LZERO WHERE C3.
C
C  CHECK RETURN CODE
C
       IF(RCODE.EQ.4) GO TO 700
       IF(RCODE.EQ.0) GO TO 39
       PRINT 35,RCODE,C3
   35  FORMAT(* RETURN CODE = *,I3,* FOR SUBSTRUCTURE *,A10)
       GO TO 900
   39  ISW=0
C
   40  CONTINUE
       ISW=ISW+1
C
C  RETRIEVE ELEMENT IDS FOR SUBSTRUCTURE FROM DATA BASE
C
*PL    GETD LONE NEXT.
```

```
C   CHECK RETURN CODE
C
       IF(RCODE.EQ.4) GO TO 49
       IF(RCODE.EQ.0) GO TO 50
       PRINT 45,RCODE,C3
   45  FORMAT(* RETURN CODE = *,I3,* FOR SUBSTRUCTURE *,A10,* AT CHKPT2*)
       GO TO 900
   49  ISW=-1
   50  CONTINUE
C
C   GET ELEMENT GRID INFORMATION
C
*PL    GETD LTWO NEXT.
C
C   CHECK RETURN CODE
C
       IF(RCODE.EQ.4) GO TO 70
       IF(RCODE.EQ.0) GO TO 60
       PRINT 55,RCODE,C5
   55  FORMAT(* RETURN CODE = *,I3,* FOR ELEMENT *,I7)
       GO TO 900
   60  I=I+1
       GRID(I)=C12
       XCORD(I)=C13
       YCORD(I)=C14
       ZCORD(I)=C15
       PSPC(I)=C16
       IF(ISW.EQ.1) GO TO 81
C
C
C  WRITE ELEMENT INFO ON UNIT 10
C
   70  WRITE(10,80) CM6,M5,M10,GRID(I-1),GRID(I),CM7,CM8,CM9
   80  FORMAT(A8,4I8,3F8.1,7X,*1*,8X)
   81  M=M+1
       CM6=C6
       M5=C5
       M10=C10
       CM7=C7
       CM8=C8
       CM9=M9
       PID(M)=C10
       IF(ISW.NE.-1) GO TO 40
C
C  REORDER GRID POINTS IN INCREASING ORDER
C
   85  DO 95 K=2,I
       LP=I-K+1
       DO 90 L=1,LP
       IF(GRID(L).LE.GRID(L+1)) GO TO 90
       ITEMP=GRID(L+1)
       GRID(L+1)=GRID(L)
       GRID(L)=ITEMP
       TEMP=XCORD(L+1)
       XCORD(L+1)=XCORD(L)
       XCORD(L)   =TEMP
       TEMP=YCORD(L+1)
       YCORD(L+1)=YCORD(L)
       YCORD(L)   = TEMP
       TEMP=ZCORD(L+1)
       ZCORD(L+1)=ZCORD(L)
       ZCORD(L)   = TEMP
       ITEMP=PSPC(L+1)
       PSPC(L+1)=PSPC(L)
       PSPC(L)=ITEMP
   90  CONTINUE
   95  CONTINUE
```

```
C
C  WRITE GRID DATE ON UNIT 10
C
      ITEMP=0
      DO 110 J=1,I
      IF(ITEMP.EQ.GRID(J)) GO TO 110
      WRITE(10,100) GRID(J),XCORD(J),PSPC(J)
 100  FORMAT(*GRID    *,I8,8X,F8.2,24X,I8,16X)
      ITEMP=GRID(J)
 110  CONTINUE
C
C  REORDER PROPERTY IDS INTO INCREASING ORDER
C
      DO 114 I = 2,M
      LP = M-I+1
      DO 111 L= 1,LP
      IF(PID(L).LE.PID(L+1)) GO TO 111
      IPID = PID(L+1)
      PID(L+1) = PID(L)
      PID(L) = IPID
  111 CONTINUE
  114 CONTINUE
C
*PL   GET1 LTHRE FIRST.
C
C  RETRIEVE ELEMENT PROPERTIES
C
      IPID = 0
C
      DO 150 I=1,M
      IF(PID(I).EQ.IPID) GO TO 150
C
      C21 = PID(I)
*PL   GET1 LTHRE WHERE C21.
C
C  CHECK RETURN CODE
C
      IF(RCODE.EQ.0) GO TO 120
      PRINT 115,RCODE,PID(I)
 115  FORMAT(* RETURN CODE = *,I3,* FOR PROPERTY ID *,I7)
      GO TO 900
C
C  WRITE MATERIAL PROPERTIES ON UNIT 10
C
 120  WRITE(10,130) C29,C21,C22,C23,C24
 130  FORMAT(A8,2I8,2A8,40X)
      WRITE(10,140) C22,C25,C27,C28
 140  FORMAT(*MAT1    *,I8,A8,8X,2F8.2,32X)
      IPID = PID(I)
 150  CONTINUE
*PL   GET1 LFIVE FIRST.
      GO TO 161
 160  CONTINUE
C
C  RETRIEVE OMITTED GRID POINTS
C
*PL   GET1 LFIVE NEXT.
C
C  CHECK RETURN CODE
C
 161  IF(RCODE.EQ.0) GO TO 170
      IF(RCODE.EQ.4) GO TO 190
      PRINT 165,RCODE
 165  FORMAT(* RETURN CODE FOR LFIVE IS *,I3)
      GO TO 900
```

MICROFILMED FROM
BEST AVAILABLE COPY

```
C
C  WRITE OMITTED COORDINATES ON UNIT 10.
C
 170  WRITE(10,180) C40,C41
 180  FORMAT(*OMIT     *,2I8,56X)
      GO TO 160
 190  CONTINUE
*PL   GETI LFOR FIRST.
      GO TO 192
 191  CONTINUE
C
C  RETRIEVE SINGLE POINT CONSTRAINTS
C
*PL   GETI LFOR NEXT.
C
C  CHECK RETURN CODE
C
 192  IF(RCODE.EQ.4) GO TO 220
      IF(RCODE.EQ.0) GO TO 200
      PRINT 195,RCODE
  195 FORMAT(* RETURN CODE FOR LFOR IS *,I3)
      GO TO 900
C
C  WRITE SINGLE POINT CONSTRAINTS ON UNIT 10
C
  200 WRITE(10,210) C36,C37,C38
  210 FORMAT(*SPC      *,3I8,48X)
      GO TO 191
  220 CONTINUE
*PL   GETI LATE FIRST.
      GO TO 224
 221  CONTINUE
C
C  RETRIEVE EIGENVALUE INFORMATION
C
*PL   GETI LATE NEXT.
C
C  CHECK RETURN CODE
C
 224  IF(RCODE.EQ.0) GO TO 230
      PRINT 225,RCODE
  225 FORMAT(* RETURN CODE FOR LATE = *,I3)
      GO TO 900
C
C  WRITE EIGENVALUE INFORMATION ON UNIT 10
C
  230 WRITE(10,240) C43,C44,C45,C46,C47,C48
  240 FORMAT(*EIGR     *,I8,A8,2F8.1,2I8,16X,*+BC*,5X)
      WRITE(10,250) C49
  250 FORMAT(*+BC      *,A8,64X)
C
C  READ REST OF DECK IN AND STORE ON UNIT 10
C
  252 READ(9,2) (DECKIN(I),I=1,6)
      IF(EOF(9))255,253
  253 WRITE(10,2) (DECKIN(I),I=1,6)
      GO TO 252
C
C  WRITE ENDDATA ON UNIT 10
C
 255  WRITE(10,260)
  260 FORMAT(*ENDDATA *,72X)
      ENDFILE 10
      REWIND 10
      GO TO 999
 700  PRINT 750,C3
 750  FORMAT(* NO SUBSTRUCTURE WITH NAME *,A10)
      GO TO 20
 800  PRINT 850
 850  FORMAT(* END OF DATA INPUT*)
C
C  WRITE FINAL ERROR MESSAGE
C
  900 PRINT 950
  950 FORMAT(//* PROGRAM TERMINATED FOR ABOVE REASON*//)
  999 STOP
      END
```

APPENDIX B

PRE-PROCESSOR PROGRAM TO CONVERT DATA IN DATA BASE TO SPAR INPUT FORMAT

```
      PROGRAM SPARS2K(INPUT,OUTPUT,TAPE9,TAPE10=OUTPUT)
      DIMENSION DECKIN(8),ISPC(6),IGRD(4)
      INTEGER C5,C10,C12,C21,C22,C16,C36,C37,C40,C41,C43,C47,C48,C18
      INTEGER SCHNME,RCODE,FILLER,PASSWD,DATASN,DATASP,TIME,DATE,CYCLE,
     1 SEPSYM,ENTRYT,STATUS
C
C  COMMON BLOCK FOR PICASSO DATA BASE
C
*PL   COMMBLOCK/PICASSO/SCHNNE,RCODE,FILLER,LASTDS,PASSWD,DATASN,DATASP,
*PL   LEVEL,TIME,DATA,CYCLE,SEPSYM,ENTRYT,STATUS.
C
C   SCHEMA FOR TOTAL STRUCTURE
C
*PL   SCHEMA/LMONE OF PICASSO/C1.
C
C  SCHEMA FOR REPEATING GROUP 2 (SUBSTRUCTURE NAMES)
C
*PL   SCHEMA/LZERO OF PICASSO/C3.
C
C  SCHEMA FOR REPEATING GROUP 4 (ELEMENTS)
C
*PL   SCHEMA/LONE OF PICASSO/C5,C6,C7,C8,C9,C10.
C
C  SCHEMA FOR REPEATING GROUP 11 (GRID POINTS)
C
*PL   SCHEMA/LTWO OF PICASSO/C12,C13,C14,C15,C16.
C
C  SCHEMA FOR REPEATING GROUP 20 (ELEMENT PROPERTIES)
C
*PL   SCHEMA/LTHRE OF PICASSO/C21,C22,C23,C24,C25,C26,C27,C28,C29.
C
C  SCHEMA FOR REPEATING GROUP 36 (SINGLE POINT CONSTRAINTS)
C
*PL   SCHEMA/LFOR OF PICASSO/C36,C37,C38.
C
C  SCHEMA FOR REPEATING GROUP 39 (OMITTED COORDINATES)
C
*PL   SCHEMA/LFIVE OF PICASSO/C40,C41.
C
C  SCHEMA FOR REPEATING GROUP 42 (EIGENVALUE INFORMATION)
C
*PL   SCHEMA/LATE OF PICASSO/C43,C44,C45,C46,C47,C48,C49.
*PL   END SCHEMAS.
C
      DATA CONMAT/10HMATERIAL C/
C
C READ IN DESCRIPTIVE INFORMATION AND CARDS THROUGH MATERIAL CONSTANTS
C
    1 READ(9,2)(DECKIN(I),I=1,8)
    2 FORMAT(8A10)
      WRITE(10,2) (DECKIN(I),I=1,8)
      IF(DECKIN(1).NE.CONMAT) GO TO 1
*PL   START S2K.
      PASSWD=6HROGERS
C
C OPEN PICASSO DATA BASE
C
*PL   OPEN PICASSO.
```

```
C CHECK RETURN CODE
C
      IF(RCODE.EQ.0) GO TO 10
      PRINT 5,RCODE
    5 FORMAT(* DATA BASE CANNOT BE OPENED - RETURN CODE = *,I3)
      GO TO 900
C
C CHECK STATUS
C
   10 IF(STATUS.EQ.0) GO TO 20
      PRINT 15,STATUS
   15 FORMAT(* DATA BASE DAMAGED - STATUS = *,I2)
      GO TO 900
C
C RETRIEVE STRUCTURE FROM DATA BASE
C
   20 CONTINUE
*PL   GETI LMONE NEXT.
C
C CHECK RETURN CODE
C
      IF(RCODE.EQ.0) GO TO 40
      PRINT 35,RCODE
   35 FORMAT(* RETURN CODE = *,I3,* FOR STRUCTURE*)
      GO TO 900
   40 CONTINUE
C RETRIEVE MATERIAL PROPERTIES FROM DATA BASE
C
*PL   GETD LTHRE NEXT.
C
C CHECK RETURN CODE
C
      IF(RCODE.EQ.0) GO TO 50
      PRINT 45,RCODE
   45 FORMAT(* RETURN CODE = *,I3,* FOR MATERIAL CONSTANTS*)
      GO TO 900
C
C WRITE MATERIAL CONSTANTS ON UNIT 10
C
   50 WRITE(10,55) C25,C27,C28
   55 FORMAT(* 1 *,A7,1X,F3.2,1X,F3.2)
C
C GET JOINT INFORMATION FROM DATA BASE
C
      WRITE(10,61)
   61 FORMAT(* JOINT LOCATIONS*)
   60 ISW=0
   62 CONTINUE
*PL   GETD LONE NEXT.
C
C CHECK RETURN CODE
C
      IF(RCODE.EQ.4) GO TO 69
      IF(RCODE.EQ.0) GO TO 70
      PRINT 65,RCODE
   65 FORMAT(* RETURN CODE FOR LONE = *,I3)
      GO TO 900
   69 ISW=-1
   70 CONTINUE
*PL   GETD LTWO NEXT.
```

```
C
C CHECK RETURN CODE
C
  73  IF(RCODE.EQ.4) GO TO 62
      IF(RCODE.EQ.0) GO TO 80
      PRINT 75,RCODE
  75  FORMAT(* RETURN CODE FOR LTWO = *,I3)
      GO TO 900
C
C WRITE JOINT LOCATIONS ON UNIT 10
C
  80  WRITE(10,85) C12,C13
  85  FORMAT(I2,1X,F3.1)
      IF(ISW.NE.-1) GO TO 62
  90  CONTINUE
C
C RETRIEVE CONSTRAINTS FROM THE DATA BASE
C
      WRITE(10,95)
  95  FORMAT(* CONSTRAINT CASE 1*)
      ISPC(1)=C16/100000
      DO 110 I = 2,6
      K=6-I
      ISPC(I)=C16/10**K
      N=I-1
      M=I
      DO 105 L = 1,N
      M=M-1
      ISPC(I)=ISPC(I)-(ISPC(L)*10**M)
 105  CONTINUE
 110  CONTINUE
      DO 111 I = 1,6
      IF(ISPC(I).NE.0) GO TO 112
 111  CONTINUE
 112  IF(I.GE.6) GO TO 120
      I=I-1
      K=6-I
      DO 115 J = 1,K
      ISPC(J)=ISPC(J+I)
 115  CONTINUE
 120  WRITE(10,125) (ISPC(J),J=1,K)
 125  FORMAT(* ZERO *,6(I1,1X))
      WRITE(10,130)
 130  FORMAT(*'XQT ELD*)
      WRITE(10,135)
 135  FORMAT(*E23*/*NSECT=1*)
 140  CONTINUE
*PL   GETD LONE FIRST.
      GO TO 143
 142  CONTINUE
C
C  GET ELEMENTS
C
*PL   GETD LONE NEXT.
```

```
C
C CHECK RETURN CODE
C
 143   IF(RCODE.EQ.4) GO TO 200
       IF(RCODE.EQ.0) GO TO 150
       PRINT 145,RCODE
 145   FORMAT(* RETURN CODE FOR LONE = *,I3)
       GO TO 900
 150   CONTINUE
C
C  GET GRID POINTS FOR ELEMENTS
C
       DO 170 I=1,4
*PL    GETD LTWO NEXT.
C
C CHECK RETURN CODE
C
       IF(RCODE.EQ.4) GO TO 180
       IF(RCODE.EQ.0) GO TO 160
       PRINT 155,RCODE
 155   FORMAT(* RETURN CODE FOR LTWO = *,I3)
       GO TO 900
 160   IGRD(I)=C12
 170   CONTINUE
C
C WRITE GRID POINTS ON UNIT 10
C
 180   I=I-1
       WRITE(10,190)(IGRD(J),J=1,I)
 190   FORMAT(4(I2,1X))
       GO TO 142
C
C WRITE EXIT
C
 200   WRITE(10,210)
 210   FORMAT(*'XQT EXIT*)
       GO TO 999
C
C ERROR EXIT
C
 900   PRINT 910
 910   FORMAT(/* JOB TERMINATED DUE TO ABOVE ERROR*)
 999   CONTINUE
C
C  CLOSE PICASSO
C
*PL    CLOSE PICASSO.
*PL    END PROCEDURE.
       STOP
       END
```

1. REPORT NO.
NASA CP-2131

2. GOVERNMENT ACCESSION NO.

3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE
Eighth NASTRAN® Users' Colloquium

5. REPORT DATE
May 1980

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)

8. PERFORMING ORGANIZATION REPORT #

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Goddard Space Flight Center
Greenbelt, Maryland 20771

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

12. SPONSORING AGENCY NAME AND ADDRESS
National Aeronautics and Space Administration
Washington, D.C. 20546

13. TYPE OF REPORT & PERIOD COVERED
Conference Publication

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This publication is comprised of papers presented at the Eighth NASTRAN Users' Colloquium held at the NASA Goddard Space Flight Center, October 30-31, 1979. The authors discuss the general application of finite element methodology and the specific application of NASTRAN to a wide variety of static and dynamic structural problems.

17. KEY WORDS
NASTRAN, structures
Finite element analysis
Substructuring
Colloquium

18. DISTRIBUTION STATEMENT
Unclassified — Unlimited

Subject Category 39

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	248	$9.50

* For sale by the National Technical Information Service, Springfield, Virginia 22161

244

NASA-Langley, 1980

END

June 18, 1981